Improvements to the alignment process in electron-beam lithography

A THESIS SUBMITTED TO
THE DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF GLASGOW
IN FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
K. E. Docherty
September 2009

Problems worthy of attack, prove their worth by hitting back.

Piet Hein.

For Pamela, Mum & Dad,
each, to whom, I owe so much.

Abstract

Electron beam lithography is capable of defining structures with sub-10 nm linewidths. To exploit this capability to produce working devices with structures defined in multiple 'lithographic steps' a process of alignment must be used. The conventional method of scanning the electron beam across simple geometrically shaped markers will be shown inherently to limit the alignment accuracy attainable. Improvements to alignment allow precise placement of elements in complex multi-level devices and may be used to realise structures which are significantly smaller than the single exposure resist limit.

Correlation based alignment has been used previously as an alignment technique, providing improvements to the attainable accuracy and noise immunity of alignment. It is well known that the marker pattern used in correlation based alignment has a strong influence on the magnitude of the improvements that can be realised. There has, to date, however, been no analytical study of how the design of marker pattern affects the correlation process and hence the alignment accuracy possible. This thesis analyses the correlation process to identify the features of marker patterns that are advantageous for correlation based alignment. Several classes of patterns have been investigated, with a range of metrics used to determine the suitability and performance of each type of pattern. Penrose tilings were selected on this basis as the most appropriate pattern type for use as markers in correlation based alignment.

A process for performing correlation based alignment has been implemented on a commercial electron beam lithography tool and the improvements to the alignment accuracy

have been demonstrated. A method of measuring alignment accuracy at the nanometer scale, based on the Fourier analysis of inter-digitated grating has been introduced.

The improvements in alignment accuracy realised have been used to facilitate the fabrication of 'nanogap' and 'nanowire' devices – structures which have application in the fields of molecular electronics and quantum conduction. Fabrication procedures for such devices are demonstrated and electrical measurements of such structures presented to show that it is a feasible method of fabrication which offers much greater flexibility than the existing methods for creating these devices.

Acknowledgements

I am indebted to a multitude of people, without whom, the work presented in this thesis would have been impossible. While it would be impossible to exhaustively thank all those who have aided me, I do want to bring special attention to these few people.

First and foremost, I wish to thank John Weaver for his supervision of this work; his constant guidance, encouragement and support has shaped this work into something greater than the sum of its parts and has been invaluable in transforming my attitudes and thought processes from those of a student into those of a researcher. His, seemingly limitless, knowledge not only enlightened my experimentation, but also has been of huge educational benefit and the numerous lively discussions ranging from politics to films, literature to science proved informative, profound and often amusing.

I would also like to thank Stephen Thoms for his helpful advice on a myriad of nanofabrication problems, the insights he was able to provide from his irreplaceable experience of electron beam lithography and for his patience when things didn't always go to plan.

I was fortunate enough to be working alongside many experienced people in the James Watt Nanofabrication Centre, and I owe much thanks to the technical staff, for their continued hard work "oiling the wheels". The experience and advice of many older and wiser lab users has been incalculably important and is due much thanks. The academic management also deserve my appreciation for the foresight and effort expelled in setting-up and running such a facility without which this work could not have been

performed.

I also owe a huge amount of gratitude to Hans Romijn of Vistec Lithography, who has given me great encouragement through his appreciation of this work. I wish to thank him for taking me under his wing and providing the opportunity to incorporate this work into Vistec's software systems. Similarly, I wish to thank Kevin Lister of the Center of MicroNanoTechnology, École Polytechnique Fédéral du Lausanne, for his work, help and perseverance, since his contribution, along with that of Hans, was critical in much of the work described in chapter 5. I also wish to thank Kevin on a personal level for his advice and guidance.

Continuing on a personal level, I wish to thank my colleagues, Alan Cumming, Steven Bentley, Dave Moran, Richard Oxland and Richard Hill, among others, for many lively lunch-time discussions. I also wish to thank the members for the coffee club who not only provided interesting diversions but suggested many interesting avenues of research. Particular mention should be made of Phil Dobson, who I am grateful to for his suggestion of investigating Penrose tilings.

Last but not least I owe a huge debt of gratitude to my family. My Dad for inspiring me as a child, and my Mum for always being there when called upon. My wife Pamela, for more than words can say. My church family, whose genuine concern is as touching as it is unfailing, and above all, Jesus, for his love that makes all worthwhile.

Contents

List of Tables

List of Figures

List of Symbols

A	Aperture pattern, 58
$A(\hat{\mathbf{r}})$	Autocorrelation, 72
D	Electron dose deposited in the resist, 23
$E\{\}$	Ensemble average, 73
E_{e^-}	Electron energy, 44
E_{tip}	Electric field strength at the tip of a field emission source, 11
G	Recovery function, 58
$H()$	Matched filter transfer function, 48
I	Image, 58
I_e	Beam current, 23
N	Noise function, 58
N_Ω	Number of pixels in Ω set, 76
N_p	Number of pixels, 104
N_x	Number of pixels in the x-direction, 96
N_y	Number of pixels in the y-direction, 96
O	Object, 58
$S()$	Fourier transform of signal function, 48
S_f	Number of frame samples, 96
S_p	Number of point samples, 96
V_0	Potential applied to upper plate of an electrostatic lens, 14
V_1	Potential applied to lower plate of an electrostatic lens, 14
V_{tip}	Voltage applied to the tip of a field emission source, 11
Z	Atomic number, 44
Ω	Set of pixels with values less than half of the peak value, 76
$\boldsymbol{B}$	Magnetic field, 13
$\boldsymbol{E}$	Electric field, 13
$\boldsymbol{F_e}$	Force on an electron passing through an electromagnetic field, 13

$\boldsymbol{v_e}$	Velocity of an electron, 13
λ	Interferometer wavelength, 18
$\hat{\mathbf{r}}$	A point in continuous real space, 48
$\hat{\mathbf{t}}$	A point in time, 48
τ_x	Arbitrary x offset, 74
τ_y	Arbitrary y offset, 74
$\tilde{O}$	Reconstructed object, 58
$a_0 \ldots a_n$	Sequence, 63
c_k	kth aperiodic autocorrelation, 63
f	Frequency of beam deflection, 23
h	Height of image, 96
$h\,()$	Matched filter input response, 48
k	Integer, 63
$n\,()$	Noise function, 48
$p\,(\hat{\mathbf{r}})$	Pattern image, 72
q_e	Charge of an electron, 13
r	Placement precision, 23
r_{tip}	Radius of curvature of the tip of a field emission source, 11
$s\,()$	Signal function, 48
t_f	Time constant of the RC filter, 96
$var\,\{\}$	Variance, 73
w	Width of image, 96
$x\,()$	Received function, 48
$y\,()$	Filtered output function, 48
AFM	Atomic force microscopy, 170
BEF	Beam error feedback, 18
BELLE	Beamwriter exposure layout for lithographic engineers - a software tool used at the University of Glasgow to set-up ebeam exposure jobs., 28
BSE	Backscattered electron, 44
BSS	Beam step size, 24
CAD	Computer aided design, 26
CAI	Coded aperture imaging, 3
CCD	Charge-coupled device, 18
CFE	Cold field emitter, 12
DAC	Digital-to-analogue converter, 23
DCM	Dichloromethane, 153

EWF	Extra wide field, 11
FIB	Focussed ion beam, 4
GDS	Graphic data system - file format of CAD files commonly used in electron beam lithography, 26
GDSII	*see GDS*, 26
HSQ	Hydrogen silsesquioxane, a flowable oxide that can be used as a negative tone electron beam resist, 30
IPA	Isopropyl alcohol, 30
MCBJ	Mechanically controllable break junction, 4
MeCN	Acetonitrile, also known as Methyl cyanide, 153
MiBK	Methyl isobutal ketone, 30
NRA	Non-redundant array, 57
PCE	Peak-to-correlation energy ratio, 76
PMMA	Poly(methyl methacrylate), a positive tone electron beam resist, 30
PN	Pseudo-random noise, 60
PRMSR	Peak-to-root mean square ratio, 76
PSR	Peak-to-sidelobe ratio, 74
RMS	Root mean square, 76
SE	Secondary electron, 44
SEM	Scanning electron beam microscopy, 4
SNR	Signal-to-noise ratio, 73
SPM	Scanning probe microscopy, 4
TFE	Thermally assisted field emitter, 11
TMAH	Tetra-methyl ammonium hydoxide, 30
TV	television, 22
UAT	Universal alignment target, 65
UHR	Ultra high resolution, 11

UHV	Ultra-high vacuum, 12
URA	Uniformly redundant array, 57
VB6	Vistec VB6 UHR EWF electron beam lithography tool, 11
VRU	Variable resolution unit, 24

1 Introduction

Electron beam lithography has been used in the fabrication of microelectronic circuits and devices for over 50 years [1]. It has the advantage of being a "maskless" form of lithography and exhibits extremely high resolution - capable of producing lines with widths of less than 10 nm [2–7]. However, as devices become increasingly smaller and more densely packed, the tolerances on all stages of nanofabrication also become more stringent. The process of alignment [8] is a crucial step in device fabrication and therefore any improvement of the current method is of profound interest, with benefits extending to all fields of semiconductor fabrication.

The limitations of the current methods of alignment preclude the direct use of lithographic techniques in the fabrication of the smallest devices. This has forced researchers to come up with novel, but in many cases complex, methods to circumnavigate the shortcomings of poor alignment [9, 10]. These and the, so called, "self-aligned" fabrication techniques severely limit the processes which can be used. A substantial improvement in the accuracy of the alignment process would allow similar structures to be fabricated using direct lithographic definition, providing much greater freedom in the fabrication process that can be used.

To overcome the limitations of conventional alignment there have been several attempts to implement an improved alignment process. These began with the attempts in 1978 to improve the design of the markers [11] and the quality of the detected backscattered electron signal [12]. While the initial research did provide an increase in accuracy, it

still relied on the same underlying method of detection and therefore was still limited by the same fundamental problems. The first alternative process to be suggested exploited techniques from the image registration and pattern recognition fields. Holburn et al. [13] suggested that an alignment algorithm based on the process of correlation could be used to perform precise alignment in electron beam lithography.

Correlation, in image processing terms, is a process that measures the similarity between two images. As such, it can be used to locate a marker by correlating a reference image of the marker with an image of the marker as exposed on the wafer with the associated distortions present. One of the main advantages of using a correlation method is that correlation is a linear and therefore analysable process, allowing an analytical study of alignment to be performed. Secondly correlation acts as a matched filter, and as will be shown in section 2.2.4.1 this optimises the signal-to-noise ratio when locating markers in images with noise.

Correlation also removes the limitation of using simple geometric shapes as markers, since locating the markers no longer relies on scanning the electron beam across pairs of parallel edges. This opens up the possibility of investigating the use of more sophisticated patterns as markers, and indeed there have been some interesting results obtained using specially designed markers and correlation based alignment [14–17]. Correlation based methods have already been used for alignment in electron beam lithography, most recently by Anderson et al. [18], and several authors have noted that the use of more sophisticated markers leads to an improvement in the accuracy [16] or to an increase in noise immunity [15, 19]. However, there has to date been no detailed study into how the design of a marker affects the attainable accuracy of an alignment process.

There are several other fields where similar optimisation criteria are required on patterns or sequences that are also used in correlation based procedures. Correlation filters have many applications in communications and data transmission, such as multiplexing functions and error checking & correcting codes [20–22]. However, one of the first appli-

cations for patterns with 'good' autocorrelation functions was suggested by Dickie [23], who proposed the use of complex aperture patterns in X-ray astronomy to improve the signal to noise ratio of weak signals. The initial idea came from the need to boost the amount of signal reaching a detector, while maintaining the high angular resolution that is provided by using a small aperture. The principle was to use many small apertures to allow more of the signal to reach the detector, however each pinhole produces its own image at the detector and a method to 'decode' the overlapping images is necessary. This can be done by deconvolving the detected signal with the aperture pattern, however this process needs the aperture pattern to be carefully chosen to produce good results. Deconvolution of the overlapping images is, in general, an "ill-posed" problem. However, choosing an aperture pattern which has a single peak in its autocorrelation and no zeros in the surrounding region allows a good approximation of the deconvolution to be performed [24].

In X-ray astronomy this technique is known as "coded aperture imaging" (CAI) [25, 26] and has been widely developed, especially in terms of optimum aperture pattern design for signal to noise ratio improvements [27, 28]. The results of these studies have suggested that important types of aperture patterns can be constructed from two classes of arrays, namely non-redundant arrays and uniformly redundant arrays [29–31]. These types of arrays have apertures arranged such that the separation between any two openings is unique for non-redundant arrays or each separation appears the same number of times for the uniformly redundant arrays. Both these types of patterns have autocorrelation functions with a single strong peak, surrounded by uniform, non-zero areas, which allows them to be used to produce good results in coded aperture imaging.

The strong autocorrelation peaks in non and uniformly redundant arrays stem from the aperiodicity of these classes of patterns. This is also a desirable property for long sequences used in certain communications systems and is particularly useful in error checking codes. One widely used set of functions for this application are maximal length

pseudo-random noise sequences produced by linear feedback shift registers [32–34]. These one-dimensional maximal length pseudo-random noise sequences, or "m-sequences", can be adapted into two-dimensional arrays which have similar, sharply single-peaked, autocorrelations with approximately uniform sidebands [35, 36].

There are therefore a wide range of patterns which exhibit properties that are advantageous for correlation or convolution based processes. There have been a variety of procedures used in the literature (see Kumar and Hassebrook [37] for a review) to assign a measure of the quality of autocorrelations; in particular a great deal of work has been done to quantify correlation filters for use in signal processing applications [38–42]. The performance criteria for correlation filters is necessarily similar to the ideal autocorrelation properties for alignment, as described in section 3.2.2, and therefore it is anticipated that the performance measures developed in this field can be directly applied to image based alignment applications.

Improvements to the alignment process simplify or even enable the fabrication of several types of generic structures that have applications across a range of devices. Two of the most interesting types of structures that could be manufactured more easily with improved alignment accuracy are "nanogaps" and "nanowires". Nanogaps are pairs (or more) of contacts positioned such that there is a gap of the order of a few nanometers between them. They have applications in the active areas of research into molecular electronics [43, 44], and to the study of electron flow through molecules [45, 46]. Traditionally, nanometer sized gaps are formed using a mechanically controllable break junction (MCBJ) [47–55] or by passing a large current through a constricted wire which forms a gap through electromigration [56–62]. Shadow masking [63], electron beam induced deposition by scanning electron microscopy (SEM) or focussed ion beam (FIB) deposition [64, 65], stress relaxation in deposited metal films [66], scanning probe microscopy (SPM) [67] and the related "Dip-pen" lithography [68] and controlled molecular beam epitaxy [69] have all also been used to form nanometer sized gaps between electrodes and allow

electron transport through molecules to be studied.

The fabrication of extremely thin metallic wires, or nanowires, has applications in studying the effects of quantum conduction and the flow of one-dimensional electrons along atom chains [70, 71]. The fabrication of nanowires has exploited many of the same techniques used to form nanogaps, in particular MCBJs and the use of SPM to "pull" nanowires from a metal surface have been widely used [72, 73]. These methods suffer from the drawback that it is impossible to produce a nanowire which can be integrated with conventional semiconductor fabrication techniques, since as soon as the force is released on the MCBJ or the substrate is removed from the scanning probe microscope the nanowire is destroyed.

The improvements to alignment accuracy and the ability to accurately form small gaps between contacting materials also has applications in forming source-drain gaps for advanced transistors. Transistor gates could then be formed in these small gaps by again exploiting the increased alignment accuracy offered. The minimum period of gratings that can be formed can also be reduced by using a two-stage aligned fabrication process. Often the capillary forces, during development or drying, limit the minimum period of lines that can be created in resist. However if every other line in the grating is fabricated in the first layer of resist they have a much larger period, reducing the capillary forces experienced and easing the fabrication. The alternate set of lines in the grating can then be fabricated in a second layer of resist, accurately aligned to the first such that the overall structure has a much shorter period than either individual layer. The fabrication of zoneplates can also be improved in a similar manner, where the odd and even zones in the zoneplate are created in two separate, aligned exposures [74].

In the following chapter the background to electron beam lithography and the fabrication processes used throughout this work will be discussed in detail. Alignment in electron beam lithography will be explained and the limitations imposed by the standard methods used to locate markers and perform alignments will be discussed. Correlation

will be introduced as a method of finding markers and the theory behind correlation will be detailed.

In chapter 3 a review of the various types of marker patterns that could be used for alignment by correlation will be undertaken, with the features, advantages and disadvantages discussed and analysed for each type of pattern, before a justification is given for selecting Penrose patterns as the optimal marker design. Chapter 4 deals with the implementation of the correlation based alignment scheme, discussing both the software that was created to perform the correlations, as well as the integration with the electron beam lithography tool. A series of experimental results are presented which illustrate the improvements that the correlation based alignment with Penrose pattern markers provides over a conventional alignment strategy.

Since the Penrose pattern markers are more intricate it is not immediately clear what effect damage to the marker caused during standard fabrication process will have on the accuracy of alignment achievable. Chapter 5 examines how Penrose pattern markers and correlation cope under non-optimal conditions where the marker pattern is degraded or damaged in some fashion. Finally, chapter 6 details both nanogap and nanowire devices that have been fabricated using processes that exploited the improvements made to the alignment accuracy. Chapter 6 also details the results of electrical measurements made on the fabricated nanogap and nanowire devices.

2 Theory And Background

2.1 Fabrication Processes

Semiconductor device fabrication is based on a sequence of "photographic" and chemical processes to manufacture structures at the micro and nano scale. The photographic process is known as lithography, a word which comes from the Greek words $\lambda\iota\theta o\sigma$ - lithos, meaning "stone" and $\gamma\rho\alpha\phi\omega$ - grapho, meaning "to write". Literally translated as "writing on stone", the lithographic printing technique, from the late 18th century, used a flat stone slab onto which oil or fat was applied to divide the slab into hydrophobic and hydrophilic regions. Ink applied to the slab would adhere to only the hydrophilic regions and when paper was brought into contact with the slab the ink would transfer to the paper producing a copy of the hydrophilic regions on the slab. In lithography in semiconductor device fabrication the 'stone' used is called a mask and contains the pattern, and the 'paper', onto which the pattern will be 'printed', is known as the substrate.

The substrate can be made from any material that can be formed into a flat plate, for example, Si, GaAs, and Quartz among others, are widely used. The substrate material is often chosen due to its electrical, or thermal properties which may allow specific types of semiconductor devices to be realised. To write on these substrates numerous lithographic techniques exist. The three techniques most commonly used are optical lithography, electron beam lithography and imprint lithography. In optical lithography a source of illumination is directed through a mask, that contains the desired pattern, on to the top

of the substrate. This is similar to how a slide projector produces an image of the slide on a projector screen. In electron beam lithography a beam of electrons is fired at the substrate surface which is mounted on a motorised stage. Carefully moving the sample and deflecting the electron beam across the whole of the substrate's surface allows the desired pattern to be traced out. With imprint lithography a deformable layer on top of the substrate is directly structured using a moulding process.

None of these processes directly affect the surface of the substrate, so to allow the pattern to be transferred into the substrate a thin layer of 'resist' is used. In the cases of optical or electron beam lithography the resist layer is sensitive to photons or electrons. This sensitivity results in a change in chemical properties of the resist at the points where it has been exposed to the radiation source, resulting in a change in dissolution rate in a given solvent which allows the pattern to be formed in the resist layer. For imprint lithography the mould causes resist to be displaced according to the pattern. This results in thickness variations in the resist layer corresponding to the desired pattern. In all cases a further process is then performed to transfer the pattern to the substrate, this will be discussed in detail in section 2.1.3.

Each lithographic technique has advantages and disadvantages; optical lithography is widely used in the semiconductor fabrication industry due to its high throughput, however the smallest features that can be formed are limited by the wavelength of the illumination source. A great amount of work over the last 30 years has reduced the wavelength of the illumination used [75] and advances in mask making techniques such as "phase-shift masks" have allowed optical lithography to produce sub-wavelength features [76–78]. Electron beam lithography is the gold standard in terms of being able to produce the highest resolution features however its extremely low throughput precludes its use in large-scale industrial manufacturing. The high resolution and flexibility of electron beam lithography has ensured that it has a particular role in research and development applications. Imprint lithography lies somewhere between optical and electron beam

lithography, both in terms of resolution with it capable of getting close to that of electron beam lithography yet retaining a throughput similar to that of optical steppers, however it suffers from the limitation that the stamp is difficult to deform and therefore there is a limitation on the alignment accuracy that can be attained.

Since this work is focused on the use of electron beam lithography the next two sections go on to describe electron beam lithography in depth, discussing the hardware that makes up an electron beam lithography system, the process used to exposure a pattern and the initial generation of the pattern data.

2.1.1 Electron Beam Lithography

Electron beam lithography has been used since the 1950s and can be performed on a wide spectrum of hardware from converted scanning electron microscopes, through to sophisticated commercial tools engineered expressly to provide the ultimate performance. A schematic diagram of the Vistec VB6 electron beam lithography system used throughout this work is shown in figure 2.1.

While this schematic is based on a Vistec VB6 Ultra High Resolution Extra Wide Field (UHR EWF) electron beam lithography system as installed and used in the James Watt Nanofabrication Centre, the key components are common to all ebeam tools. The key systems are, the gun where electrons are generated, the column which focusses the electron beam and provides a method of controllable deflecting the beam across the substrate, the stage on which the substrate coated with resist is mounted, along with the vacuum, electronic control and power supply systems. Dedicated ebeam tools contain a pattern generation system to convert computer-aided designed patterns into control signals for the beam deflection and blanking systems, while simpler equipment, such as converted SEMs, may have an external pattern generator system. More advanced tools may also contain substrate handling systems allowing multiple substrates to be automatically loaded and unloaded from the machine for exposure in a sequential or 'batch' fashion. The entire machine may be mounted on a plinth and/or contain a

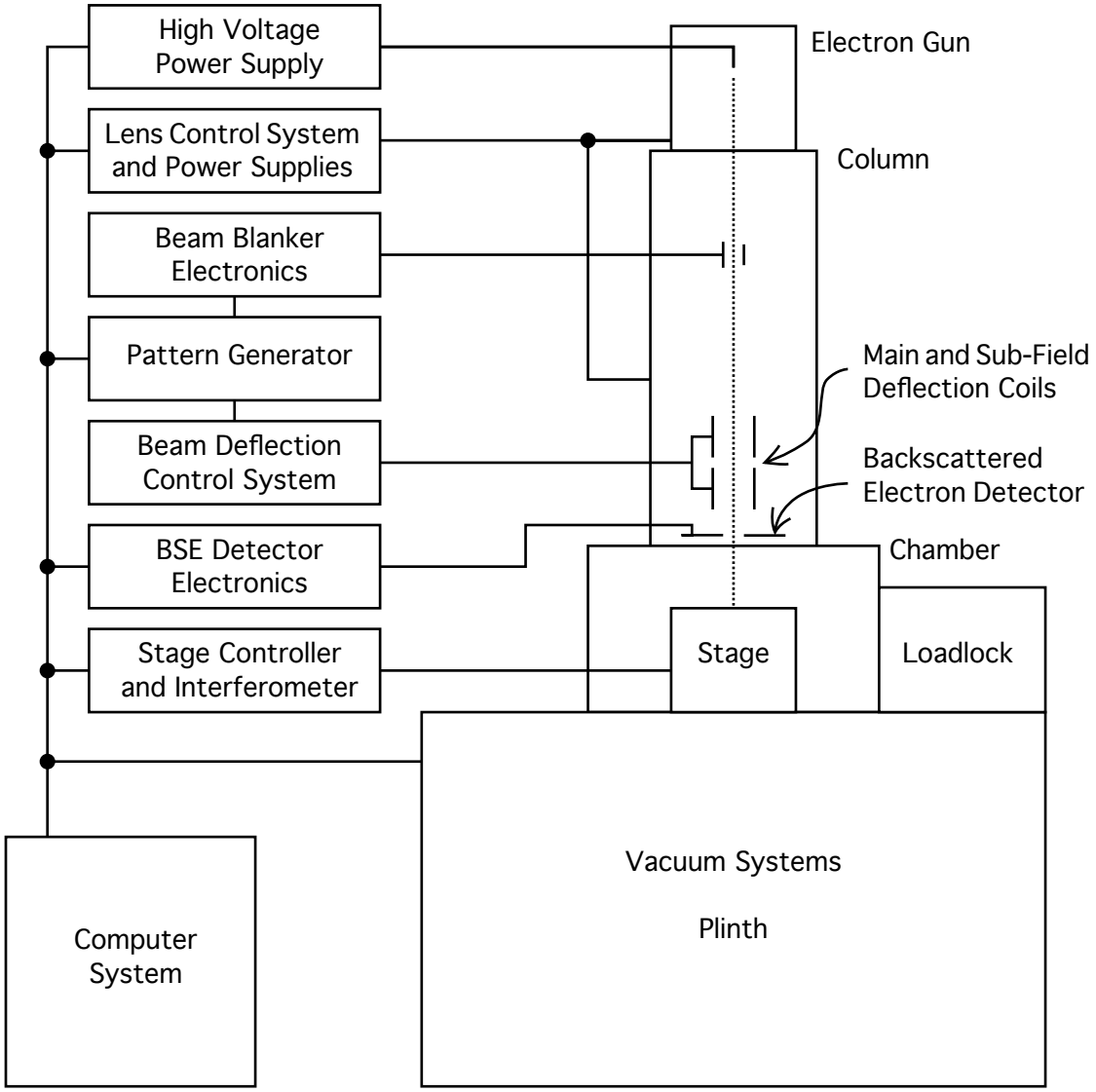


Figure 2.1: Schematic representation of a Vistec VB6 UHR EWF electron beam lithography system, showing the key components comprising an ebeam tool.

vibration isolation table, which stabilises the system reducing effects of seismic activity and isolating the tool from environmental sources of mechanical vibration.

2.1.1.1 Electron Gun

The electron gun generates the beam of electrons that is used to write the pattern on the substrate. This is achieved by a cathode emission process and the VB6 has a thermally assisted field emission source (TFE). Field emission sources work by using the fact that the electric field strength is considerably increased at sharp points. Equation 2.1 gives the electric field strength E_{tip}, when a potential V_{tip}, is applied to a point with radius r_{tip}.

$$E_{tip} = \frac{V_{tip}}{r_{tip}} \tag{2.1}$$

It can be seen that reducing r_{tip}, by sharpening the tip, causes the electric field strength for a given applied potential to increase at the tip. This increased electric field reduces the work function barrier and, if this potential is sufficiently large, or the tip sufficiently sharp, allows electrons to tunnel out of the tip forming an electron beam. For field emission to be a viable electron source the tip has to be kept perfectly clean, free from all contaminants and any oxide. This is because any atoms or molecules absorbed onto the tip will cause fluctuations in the work function of the surface of the tip and therefore also in the emission current. One method to minimise the current fluctuations is to operate under ultra-high vacuum (UHV) conditions, $< 10^{-8}$ Pa, and in this mode the electron generation process is known as "cold field emission" (CFE). Alternatively heating the tip allows the cleanliness to be maintained by increasing the evaporation rate of surface contaminants. This puts less stringent requirements on the vacuum level, around 10^{-7} Pa. The thermal energy also reduces the barrier through which the electrons tunnel, this is known as "thermal field emission" [79]. In the VB6 the tip itself is formed from a single crystal of tungsten formed into a fine point and coated in ZrO_2 which helps

to improve the stability of the source by reducing the work function of the [100] surface planes of the tungsten crystal [80]. The addition of the ZrO_2 coating turns the thermal field emitter into a "Schottky emitter". A schematic of a TFE electron gun is shown in figure 2.2.

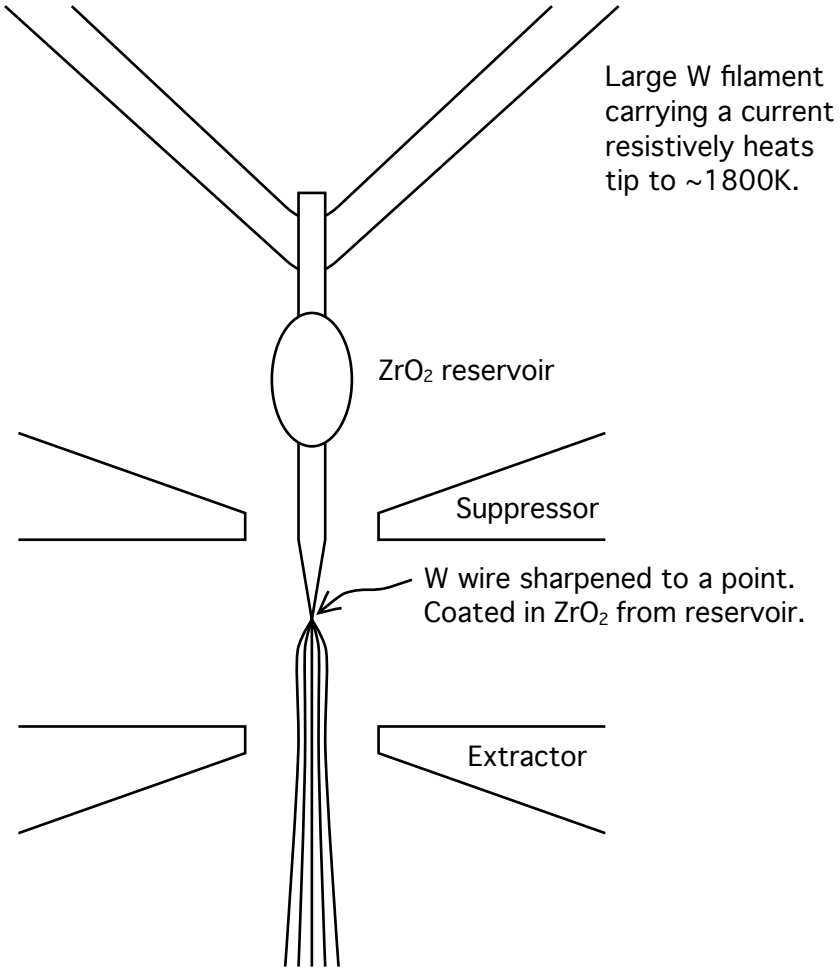


Figure 2.2: Schematic representation of a thermal field emission electron gun.

The schematic shows the sharpened tungsten tip attached to a larger tungsten filament which is used as a resistive heater to raise the temperature of the tip to around 1800 K. The ZrO_2 coating is maintained by the ZrO_2 reservoir bead part-way along the tip which replenishes the coating as it evaporates or is sputtered off the tip. The suppres-

sor electrode is held at a slightly negative potential with respect to the cathode tip and its proximity to the shank of the tungsten needle stops all but the very tip of the needle from emitting electrons. A few kV positive potential between the extractor and the cathode creates the large electric field necessary to extract electrons from the tungsten surface. Once extracted from the tungsten surface the electrons are accelerated towards the extractor, and pass through a small aperture in the extractor. The electrons are further accelerated by the anode which is held at the full accelerating potential, typically +100 kV for the VB6, and pass down the column.

2.1.1.2 Electron Lenses

The suppressor and extractor forming part of the electron gun are the first electron lens that alters the trajectory of the electron beam. In general, electron lenses produce an electromagnetic field which produces a Lorentz force, $\boldsymbol{F_e}$, on the electrons in the beam according to equation 2.2, where $\boldsymbol{E}$ and $\boldsymbol{B}$ are the electric and magnetic fields, q_e is the charge of an electron and $\boldsymbol{v_e}$ is the instantaneous velocity of the electron. The resulting force alters the trajectory of the electrons as they pass through the electromagnetic field and through careful design allows the electron beam to be focussed and deflected as it passes down the column.

$$\boldsymbol{F_e} = q_e \left(\boldsymbol{E} + \boldsymbol{v_e} \times \boldsymbol{B}\right) \tag{2.2}$$

Electron lenses can produce either an electric field, an "Electrostatic lens", or a magnetic field, an "Electromagnetic lens". Electrostatic lenses, as shown in figure 2.3, affect the electron beam by applying an electric charge to pairs of conductive plates close to the electron beam. A negative charge on the top plate bends the beam away from them while a positive charge on the lower plate attracts the electrons. The increase in momentum as the electrons pass through the lens means that the top plate has a greater bending effect than the bottom and the electron beam is forced to converge. In this way

sets of plates can be used to focus the beam as it passes down the column.

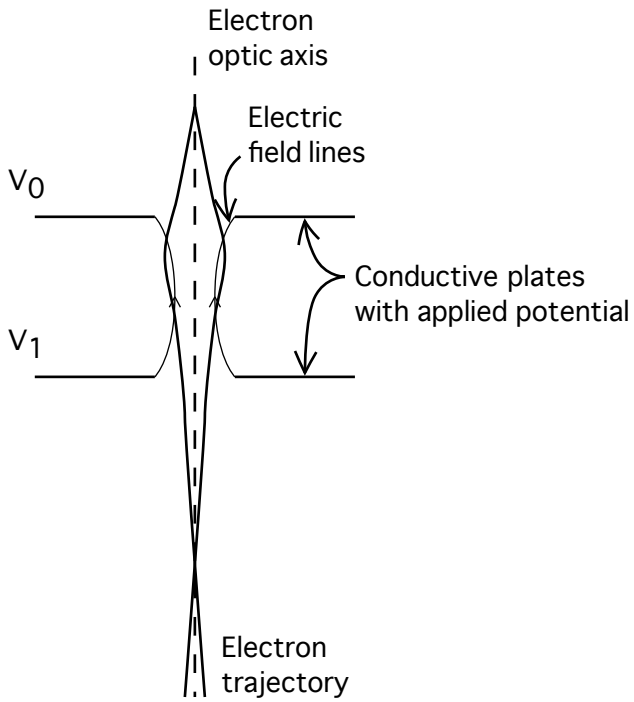


Figure 2.3: Example of an electrostatic lens showing some electric field lines and the focussing effect on an electron beam.

In an electromagnetic lens, as shown in figure 2.4, a current is passed through the windings which are contained within an iron pole piece. The flowing current produces a magnetic field in the small gap in the pole piece which then produces a force on the electrons as they travel through the field. The magnetic field is weakest on the optical axis and so the lens has a stronger effect on electrons further from the optical axis. This results in the lens acting to focus the beam as a convex lens would in an optical system. Since the Lorentz force an electron experiences is the result of the cross product of its velocity and the magnetic field there is also a rotation about the optical axis. The strength of the magnetic field can be altered by adjusting the current flowing in the windings which allows the focal length of the lens to be adjusted

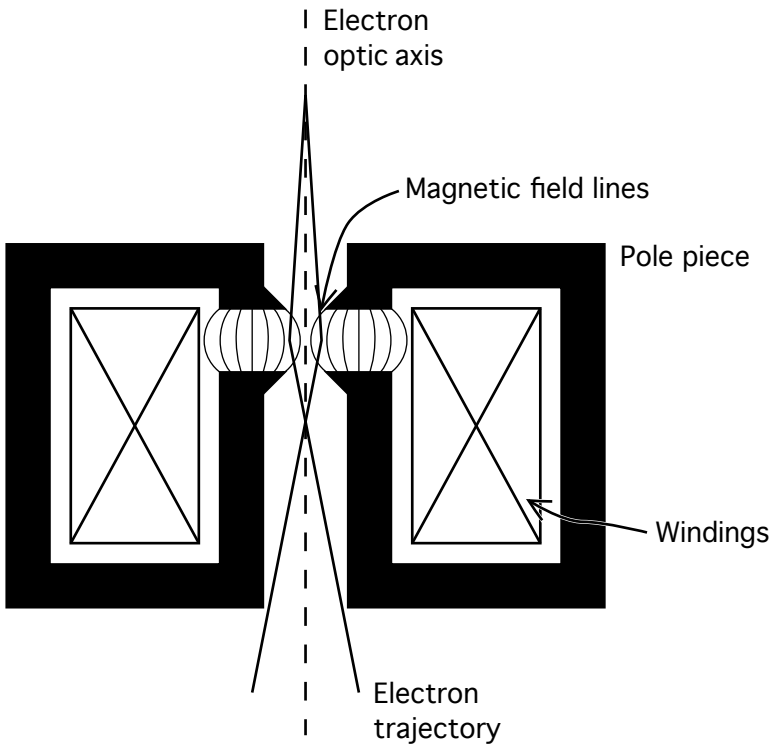


Figure 2.4: Example of an electromagnetic lens showing the magnetic field lines in the gap in the pole piece and the effect this field has on an electron beam.

2.1.1.3 Beam Blanker

The electron lenses are one of the fundamental components of the column with the others being the beam blanker, the apertures and the deflection coils. The blanker and deflection coils produce an electromagnetic field perpendicular to the optical axis. Therefore they tilt the electron beam and can be combined to provide translational shifts to the trajectory of the electron beam while not affecting its focus. As for lenses, the transverse field can be created either electrostaticaly or electromagneticaly. Electrostatic deflection systems have the advantage of very high speed and so the beam blanker uses an electrostatic deflection system, the main and sub-field deflection systems, however, are electromagnetic.

The beam blanker is a set of electrostatic plates that allow the beam to be switched

on and off very rapidly. This is achieved by deflecting the beam through a relatively large angle such that the beam strikes one of the limiting apertures within the column instead of passing through the aperture and continuing down the column. It is carefully positioned in the column so that it is centred at a point where the beam converges to a focus, also known as a beam crossover. This means that the position of the beam at the substrate plane is independent of the voltage applied to the beam blanker plates up to the point when the beam is turned off, a technique known as "conjugate plane blanking". This avoids any unintended exposure of the resist caused by a shift in the beam position as the blanker turns the beam on or off.

2.1.1.4 Apertures

The beam passes through a series of small holes, or apertures as it progresses down the column. These have three main functions. Firstly, they act to block out electrons that are a long way from the optic axis and to block any stray electrons produced in the column from reaching the substrate. These "spray apertures" have no significant effect on the shape of the beam itself. Secondly, "blanking apertures" are used to block the entire beam, as used within the beam blanker. Thirdly, "beam limiting apertures" set the convergence angle of the beam that passes down the column (the convergence angle is the half-angle of the beam as measured at the substrate surface). The two effects of this are to control the effect of lens aberrations and hence the minimum spot size that can be formed and to allow the amount of current within the beam to be set.

2.1.1.5 Deflection Coils

The lenses in the column allow the beam to be focussed precisely on the substrate's surface, and along with the apertures allow the spot size at the substrate's surface and the current carried by the beam to be controlled. The final important parts of the column are the main field and sub-field deflection coils. Like the beam blanker the deflection coils produce a transverse electromagnetic field which can be used to deflect the beam across

the surface of the substrate. Unlike the beam blanker the main and subfield deflection systems are based on coils and so produce a magnetic field. It is these coils that are used to deflect the electron beam across the substrate in the desired pattern and perform the writing.

Electron optics suffers from many sources of aberrations such as spherical aberration, coma, and chromatic aberration. Each lens in the optical system adds to the total aberration, however, the magnitude of these aberrations within a lens increases with the radial distance from the optical axis. It is therefore desirable to not deflect the beam a long way off-axis. The extent of the maximum allowed deflection is called the "main field" and this deflection is provided by the main field deflection coils. The second reason for limiting the extent of any beam deflection is because as the distance the beam is deflected increases so too does the time required for the correct current to be applied to the deflection coils and for the magnetic field to stabilise. Even so the relatively large distance that the beam can be deflected across and the associated settling time makes the main field deflection system prohibitively slow to use to actually write the pattern. To overcome this limitation the main field is then split into smaller segments called subfields. A second set of deflection coils is used to deflect the beam within a subfield and, since this is a smaller deflection, and the coils have smaller inductance, these are much faster. On the VB6 there are up to 64x64 subfields making up one main field, the main field deflection system is used to deflect the beam to the centre of one of the subfields, and the faster subfield deflection coils are then used to scan the beam over the exact points within the subfield to write the pattern.

The maximum main field size on the VB6 is 1.31072 mm, but there are obviously times when it is necessary to write a pattern which is larger than this value. To write large structures the substrate is moved on an interferometrically controlled stage. The linear motors that drive the stage on the VB6 are not accurate enough to position the stage correctly by dead-reckoning so a two-axis interferometer constantly measures the

exact position of the stage and applies an offset to the beam so that it is directed towards the correct point on the substrate, even if the stage is slightly mis-positioned. This is called the beam error feedback or BEF. The BEF can also apply a correction for drifts in the stage position over time, however the precision of this is limited to what the interferometers can detect. On the VB6 the stage has $\lambda/1024$ interferometers and so can measure the position of the stage with a resolution of 0.62 nm.

The substrate is mounted on a holder which is loaded onto the stage using kinematic mounts. This allows the holder to be accurately positioned repeatedly at the same point with respect to the stage. Various types of holder are used depending on the size and shape of the substrate being used. The holders range from 2 to 6 inch wafer holders, mask plate holders and "piece part" holders for small substrates.

2.1.1.6 Height meter

Semiconductor substrates are manufactured in a variety of standard thicknesses depending on the fragility of the substrate material and the purpose for which it will be used. When mounted on the holder the distance between the substrate's surface and the final lens within the column must be accurately measured so that the electron beam can be brought into focus at the correct point. The position of the surface of the substrate is frequently referred to as the substrate's height and is measured within the ebeam tool by a laser "height meter". The height meter, schematically shown in figure 2.5, reflects an infrared laser beam off the surface of the substrate which is then detected by a charge-coupled device or CCD array sensor. Changes in the position of the substrate's surface will alter the point on the array sensor that the reflected beam hits and this change in the detected signal can be used to calculate the separation between the final lens and the substrate.

The height meter allows the electron beam to be focussed on the surface of the substrate accurately. While this is important it is less critical than with an optical stepper for instance because electron beam lithography systems have a much greater

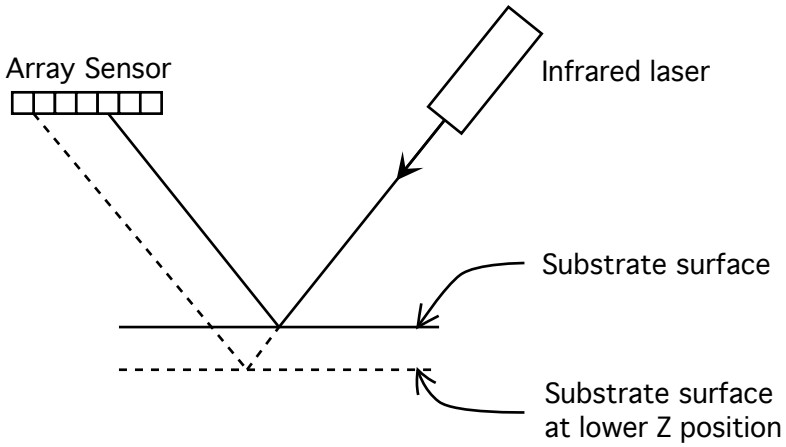


Figure 2.5: Schematic illustration of the height meter used to measure the separation between the final lens and the substrate's surface, showing how the position of the reflected laser beam on the array sensor is dependant on the height of the substrate's surface.

depth of focus, but knowing the substrate's height also allows other corrections to be applied. Examining figure 2.6 shows how the distance between the final lens and the writing surface causes a change in the apparent field size at the surface of the substrate. In the VB6 ebeam system the distance from the principle plane of the final lens to the substrate is approximately 35 mm, and the maximum main field is 1.31072 mm across. This means that there is a change in the field size equal to ∼4% of any change in the height of the substrate's surface. Therefore a change in height of 10 μm results in a change in field size of 0.4 μm (the 1.31072 mm field becomes1.31109 mm or 1.31035 mm depending on whether the surface is further away from, or closer to the final lens). This is, in effect, a change in scale of the projected pattern at the substrate's surface, but by measuring the distance of the substrate from the final lens a correction can be applied in the electron optics such that the pattern is correctly formed.

The whole of the machine is under electronic control which maintains the appropriate vacuum levels, controls the bias voltages for the gun and controls the currents and biases for the electromagnetic and electrostatic lenses and coils. The beam blanker and

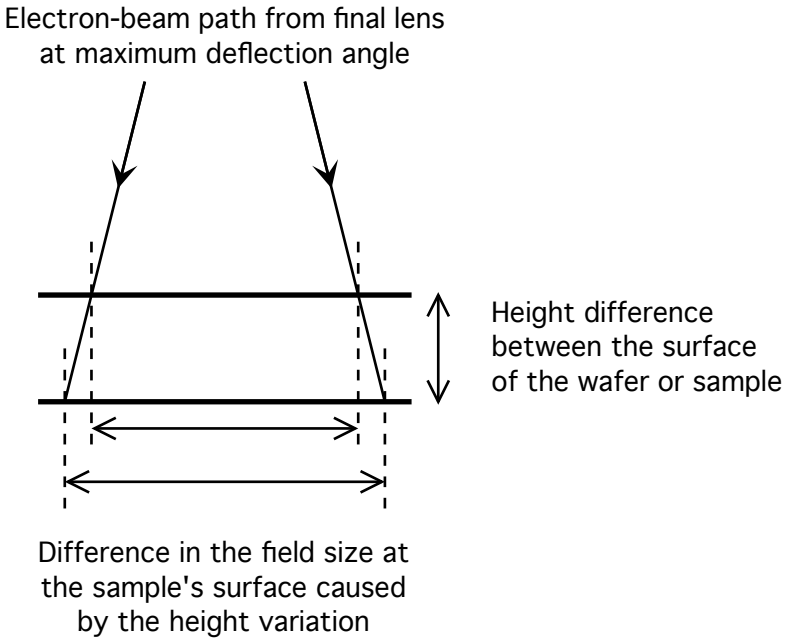


Figure 2.6: Illustration of the change in fieldsize resulting from a change in substrate height.

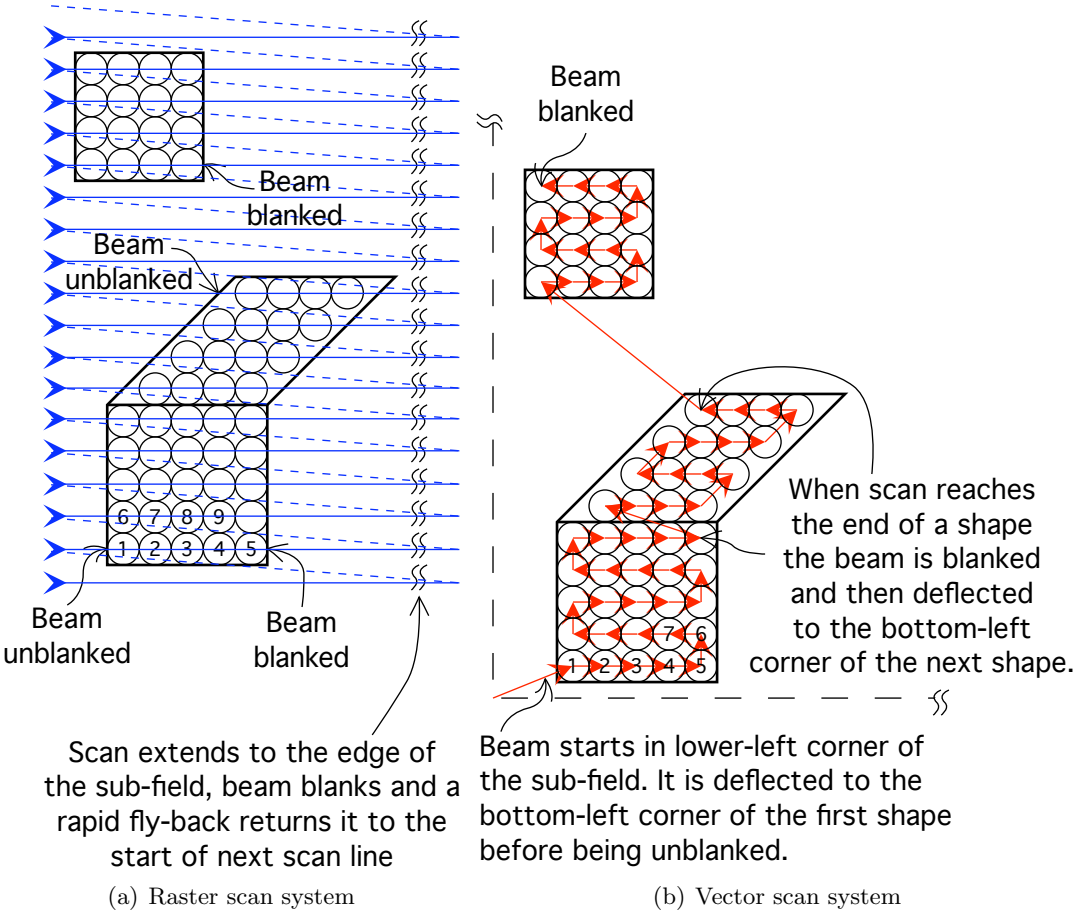
deflection coils are controlled by the pattern generator to allow the pattern to be written according to the data within the pattern file.

2.1.1.7 Classification of Ebeam Systems

Electron beam lithography systems can be classified according to both beam shape and beam deflection strategy. An important distinction between different types of electron beam lithography tools is that of shaped-beam versus Gaussian-beam. With a Gaussian beam system the electron beam is focussed down to have as small a diameter as the beam current will allow and the cross-sectional intensity profile of such a beam can be approximated by a Gaussian function. Shaped-beam systems form the electron beam into a wide beam of uniform intensity. This wide beam is then passed through a series of interchangeable apertures which allow the beam to be directly formed into trapezium shapes. In this way while with a Gaussian-beam system each shape is formed from a series of point exposures, in a shaped-beam system each shape can be exposed as one

"shot" or exposure. As a result throughput is vastly superior with shaped-beam systems when large patterns are exposed. However, this comes at the cost of ultimate resolution. Shaped-beam systems have found significant use in advanced manufacturing facilities [81–83] but their relative expense compared with Gaussian-beam systems and their slightly lower resolution capabilities means that Gaussian-beam systems are of more interest in a university research setting. The remainder of the discussion of electron beam lithography focusses on Gaussian-beam systems.

In electron beam lithography, two strategies are used to control the beam deflection and blanking; raster scanning and vector scanning, as illustrated in figure 2.7.



(a) Raster scan system (b) Vector scan system

Figure 2.7: Comparison between the beam deflection path in a raster and a vector scan system exposing the same pattern data.

With raster scanning the beam is scanned in a series of parallel scan lines across the complete subfield and the beam blanker is switched on and off to control which parts are written, figure 2.7(a). This is similar to how a black & white TV forms a picture on its screen. In a vector scan system, figure 2.7(b), the beam is deflected to the start of each shape to be written and then is scanned across that shape, before being deflected to the start of the next shape. It means that the beam is on for a much greater proportion of the writing time since it is only blanked when moving between shapes. Raster scan systems can deflect the beam at a higher rate than vector scan systems but even with high density patterns there is wasted time while areas that are not being patterned are scanned over. The time required to write with a raster scan system is independent of the pattern itself, improvements can only be made by speeding up the deflection system. With a vector scan system the write time is dependant on the pattern data. Both tools are ultimately limited by the brightness of the electron source. The VB6 ebeam tool used throughout this work is a vector scan system.

The "placement precision" of an electron beam system describes the precision with which a particular shape can be placed within the main field. The beam deflection is controlled by analogue drive signals applied to the deflection coils. These in turn are derived from digital-to-analogue converters (DAC) in the pattern generator corresponding to the pattern data being processed. The precision of these DACs controls the precision with which the beam can be deflected and hence the precision of shape placement. The VB6 uses 20-bit DACs which means that $2^{20} = 1048576$ distinct positions along each of the x and y axes can be addressed. So with the maximum field size of 1.31072 mm this corresponds to a placement precision of 1.25 nm. The VB6 can however reduce the maximum field size to provide a finer placement precision, down to a minimum field size of 0.131072 mm corresponding to a placement precision of 0.125 nm. This is very much less than any feature that could be formed in resist and although placement precisions this small have uses, for instance, accurately defining the period of gratings, these are

very special circumstances and occur very infrequently. Furthermore the ebeam tool must under go a period of calibration when defining a field size and this requires about 3 hours to perform. These "full calibrations" must be checked and rerun periodically and so a compromise between choice of field sizes and calibration time required has been made. Therefore the VB6 at the University of Glasgow is set-up to have three field sizes and corresponding placement precisions; field sizes of 1.31072 mm, 1.048576 mm and 0.524288 mm corresponding to placement precisions of 1.25 nm, 1 nm and 0.5 nm respectively are used.

The placement precision, as well as determining the accuracy with which a point can be selected, effectively also defines the smallest beam deflection that can be produced. Therefore the beam position is changed by the placement precision when the least-significant bit of the DACs is changed. The speed of this movement combined with the number of electrons in the beam define the dose deposited in a given area of the resist. To accurately measure the beam current a Faraday cup is used as illustrated in figure 2.8.

Essentially, this is formed with a small aperture above a large conductive bucket. When the beam is directed through the aperture the electron beam strikes the bottom surface of the bucket, some of the electrons are collected immediately by the metal, but some are scattered away from the surface. The bucket is designed to collect the vast majority of the scattered electrons and by measuring the current flow away from the bucket a precise measure of the beam current is obtained.

It takes a certain number of electrons to "expose" a given area of the resist and this is commonly defined as the dose. Expressed in $\mu\mathrm{Ccm}^{-2}$, it may be calculated from the beam current, I_e, the frequency, f, with which the beam is moved and the area exposed at each point, expressed as the square of the distance the beam is deflected between shots r^2, as shown in equation 2.3.

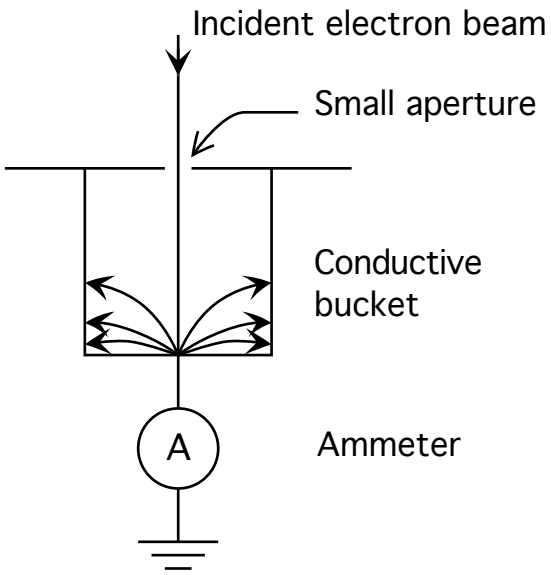


Figure 2.8: Schematic illustration of a Faraday cup used to measure the current of an electron beam, showing how all the electrons within the beam and those produced by scattering from the Faraday cup material are captured and measured by the ammeter.

$$D = \frac{I_e}{f\,r^2} \tag{2.3}$$

The lowest dose that can be used is limited by the maximum frequency, minimum current and largest beam deflection unit. Since the maximum frequency is fundamentally limited by the speed of the deflection system it cannot be increased to allow lower doses to be achieved, therefore either the current must be lowered or the area exposed at each point increased.

Since the placement precision is very small, a second grid can be used during the exposure, allowing the beam to be stepped out using a larger spacing but maintaining the positional accuracy that comes with a smaller placement precision. The ratio between these two grid is described as the "variable resolution unit", or VRU, and can be any integer between 1 and 512. The resulting grid is sometimes referred to as the beam step size, or BSS, since this is the size of the steps that the beam will be deflected across.

Mathematically the two are related according to equation 2.4.

$$\text{BSS} = \text{VRU} \times \text{Placement precision} \tag{2.4}$$

A larger BSS means that at a given frequency, lower doses can be produced, or that at a given dose, lower frequencies can be used. Figure 2.9 shows an example placement precision grid with circles showing a 3x3 array of beam exposure positions when using a VRU of 1, 2 and 4

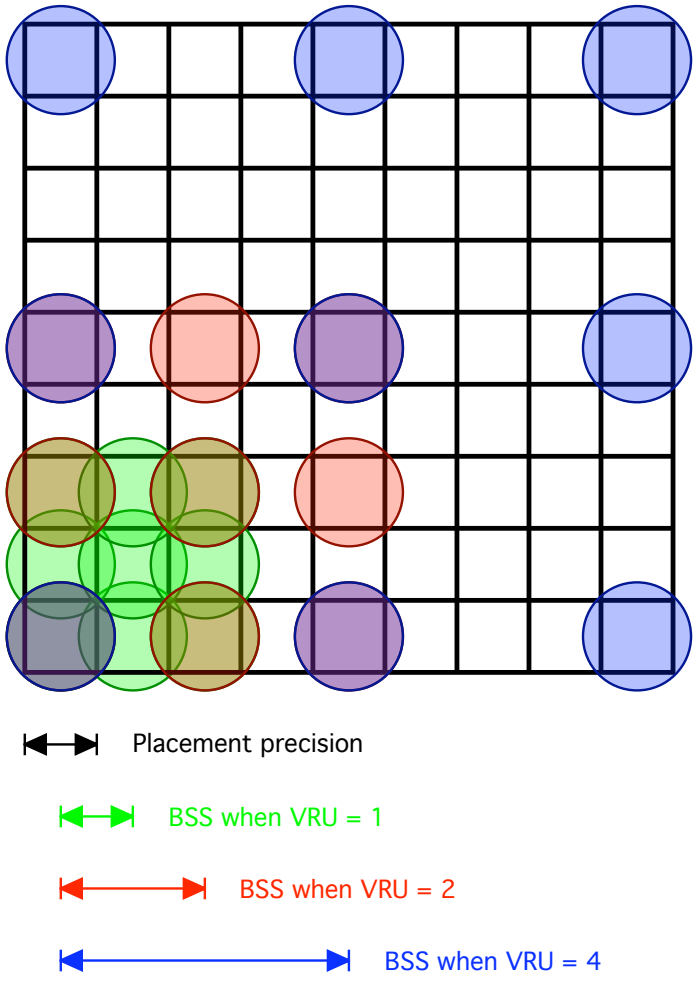


Figure 2.9: Placement precision grid with circles showing a 3x3 array of exposure points when using a beam step size of 1, 2 and 4 times the placement precision.

2.1.1.8 Pattern Design and Data Preparation

The whole aim of lithography is to form a pre-defined pattern on a physical object. To create the pattern in the first place it is extremely common to use a form of computer aided design (CAD) to create an electronic file containing the required arrangement of shapes. The predominant format for these files at Glasgow is the GDSII format, now owned by Cadence it was originally developed by Calma and derives its name from their layout design software, "Graphic Data System". It is a binary file format that can represent shapes, such as polygons, lines, circles and text in any combination, in a hierarchical manner, such that a section of the layout can be drawn once and then reused as many times as necessary in other parts of the pattern. It also allows other information about the layout to be stored, such as layer numbers and datatype which are often used to relate to the physical layers of material that will be formed.

As hinted at in the previous section, vector scan ebeam tools can only deflect their beam to expose simple shapes. In fact the VB6 can only expose trapezium shaped elements to form the pattern. It would be extremely time consuming and error prone if this limitation were extended to the pattern design process, and indeed designers can make use of CAD systems to produce complex patterns with many structures of all shapes and sizes. This is possible because there are sophisticated software tools that allow complex CAD patterns to be split into trapezium shaped elements that an ebeam tool can expose. This process is referred to as "fracturing" since the complex polygons are split-up, or fractured, into trapezium shaped elements. Fracturing not only splits large polygons up into trapezia because of their complexity but also sorts the data into fields and subfields, corresponding to the areas that can be scanned with each deflection system.

There is another important function that the fracturing software performs. That is, it fits the design data onto the placement precision grid that will be used during exposure. This is done by snapping the vertices of the pattern data to points on the placement

precision grid, such that the trapezium can be exposed by the ebeam tool. A similar process occurs during the exposure where the fractured data, already snapped to the placement precision grid, is fitted onto the beam step size grid. The placement precision grid is used to position the bottom left corner of each trapezium and the width and height are then altered since an integer number of beam steps must be used to expose each shape. Rather than being done off-line by the fracturing software, the beam-step size fitting is done on-the-fly in the pattern generator allowing the flexibility to chose the VRU used at the point of exposure. This may be needed, for instance, to expose the same pattern in two resists with different electron sensitivities and therefore requiring different doses. Since the beam step size can be relatively large, simply snapping the width and height to the nearest beam step multiple can introduce significant errors in the size of the exposed features and so sophisticated algorithms are used to minimise the errors when the vertices of the fractured data and the BSS are not congruent.

2.1.1.9 Job Layout

Once the pattern has been designed and converted into the appropriate data format via fracturing there is one last step to set up an exposure. The desired exposure may require that a single pattern file be exposed in multiple locations on a substrate, or that several different pattern files are combined to create one larger exposure. The process of defining which pattern files are to be exposed in which position is usually described as creating a job layout.

The layout contains the locations of all the pattern files to be exposed on a substrate, as well as the exposure parameters, such as dose, VRU and beam current to be used during the exposure of each pattern. At the University of Glasgow a propriety software tool called BELLE (Beamwriter Exposure Layout for Lithographic Engineers) [84] is used to create exposure layouts. This collects information about all the patterns that have been fractured via a centrally stored database and allows the patterns to be positioned on a schematic of the substrate. The exposure parameters for each pattern file can then

be configured. It uses a hierarchical system to allow arrays of patterns, or of groups of patterns, to be created and this allows very complex layouts to be created with ease. BELLE also allows alignment jobs to be set-up by defining the position of each of the markers and also the parameters to use during the marker search routines.

BELLE produces a text "job file" that contains an ordered list of the job settings, pattern positions, exposure parameters, marker positions, marker search parameters, and various other commands to allow the job to be exposed properly. At the time of exposure this file is read by the "Runjob" program [85] (see section 4.1) which controls the VB6 to perform the required calibrations, alignments and exposures.

2.1.2 Resist

All conventional forms of lithography rely on the use of a layer of resist (FIB and dip pen lithography, for example, do not), which is a layer of material which can be selectively removed so that it protects some areas of the substrate and exposes other areas to subsequent processing, for example etching. In electron beam lithography the resist is formed from a compound that undergoes chemical changes when exposed to energetic electrons. There are two forms that these changes can take and this determines the type of resist, positive or negative. When electrons interact with a "positive" resist they cause scissions in the polymer chains that make up the resist. This makes the exposed regions *more* soluble in a chemical solution known as developer. With "negative" resist, the incident electrons cause the molecular chains to cross-link [86]. This makes the exposed regions *less* soluble in the developer. Figure 2.10 shows diagrammatically the final resist profile for both positive and negative resists.

The degree of change caused by the electrons impacting on the resist is affected by the number of electrons, or the electron dose, and on the accelerating voltage, since this controls the production of secondary electrons which actually expose the resist. Similarly the rate of dissolution and the difference in final resist thickness can be controlled by varying the concentration of the developer and the amount of time used for development.

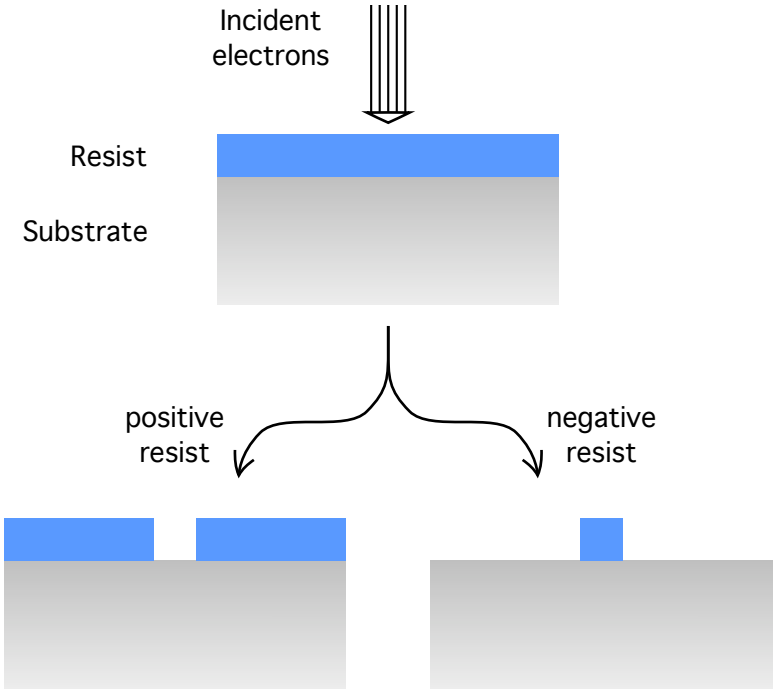


Figure 2.10: Diagram showing the differences in the final resist profile for positive and negative resist exposed to the same electron beam pattern.

In this way a set of process parameters, sometimes referred to simply as a "process" can be defined to give well defined features.

This work made use of two commonly used ebeam resists, PMMA and HSQ.

Poly(methyl methacrylate), or PMMA, has been used as a resist for electron beam lithography for nearly 40 years [87]. It is a positive resist at moderate electron doses but it can also be used as a negative resist at much higher electron doses [88, 89]. A range of chemical solutions can be used to develop PMMA however for this work the PMMA was developed in a solution of methyl isobutal ketone (MiBK) and isopropyl alcohol (IPA). PMMA has a resolution for isolated lines of much better than 20 nm, is easy to remove in acetone and is ideal for use in research environments, although it has poor dry etch resistance.

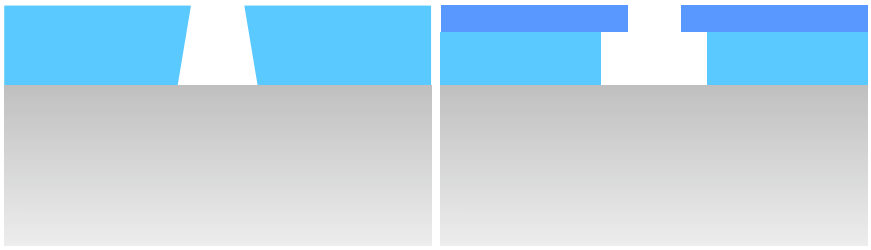
Hydrogen silsesquioxane, or HSQ, is a flowable silicon oxide which is also electron sensitive. It is a high resolution negative resist capable of producing features with sub-10 nm linewidths. It can be developed in tetra-methyl ammonium hydroxide (TMAH) or a TMAH based developer solution such as Microposit MF CD-26 developer. One interesting property of HSQ is that after exposure it forms a layer that is physically and chemically very similar to Silicon Dioxide (SiO_2). HSQ has been widely studied due to its very high resolution potential [3–6, 90] and sub-10nm half-pitch gratings have been reported in HSQ [91].

2.1.3 Pattern Transfer

There are two main fabrication processes used frequently in small scale semiconductor fabrication. The processes can be thought of as roughly inverses of each other as one of the processes is an *additive* process, "lift-off", with the other is a *subtractive* process, "etching".

The first, lift-off, allows patterned metal to be added to a substrate. It relies on the resist having an undercut profile after development such that the top of the features in the resist are slightly narrower than the bottom. This is illustrated in figure 2.11(a) where

the effect has been exaggerated for clarity. The production of such an overhang can be problematic, particularly with positive resists, and frequently resist bi-layers are used to enhance the overhang. Bi-layers use two layers of resist, one on top of the other, with the lower layer being more sensitive to the electron exposure. For PMMA this can be done by using solutions of the polymer with different molecular weights, the top layer having the larger weight and hence less sensitivity to electrons. In this way, after development the top layer of resist overhangs the lower layer as shown in figure 2.11(b).



(a) Lift-off profile in a layer of resist (b) Lift-off profile in a resist bi-layer

Figure 2.11: Diagram showing two possible resist profiles suitable for lift-off, and the way in which a resist bi-layer can aid the production of a resist overhang.

The undercut resist profile means that when a thin layer of metal is deposited on top of the whole substrate the metal becomes discontinuous at all the edges of the pattern. This relies on the metal being deposited from a collimated source perpendicular to the resist surface and relies on the metal layer being substantially thinner than the resist. Submerging the substrate in a solution which dissolves the resist removes the remaining resist along with any metal on top of it, resulting in a metal layer attached to the substrate that matches the pattern of the electron beam exposure. This process is illustrated in figure 2.12.

The metal can be deposited in a number of ways, filament evaporation, electron beam evaporation, effusion cell evaporation, and laser ablation, sputter coated metal generally cannot be used for a lift-off process. The work reported here made use of an electron beam evaporator which relies on a simple electron source to heat a crucible filled with the

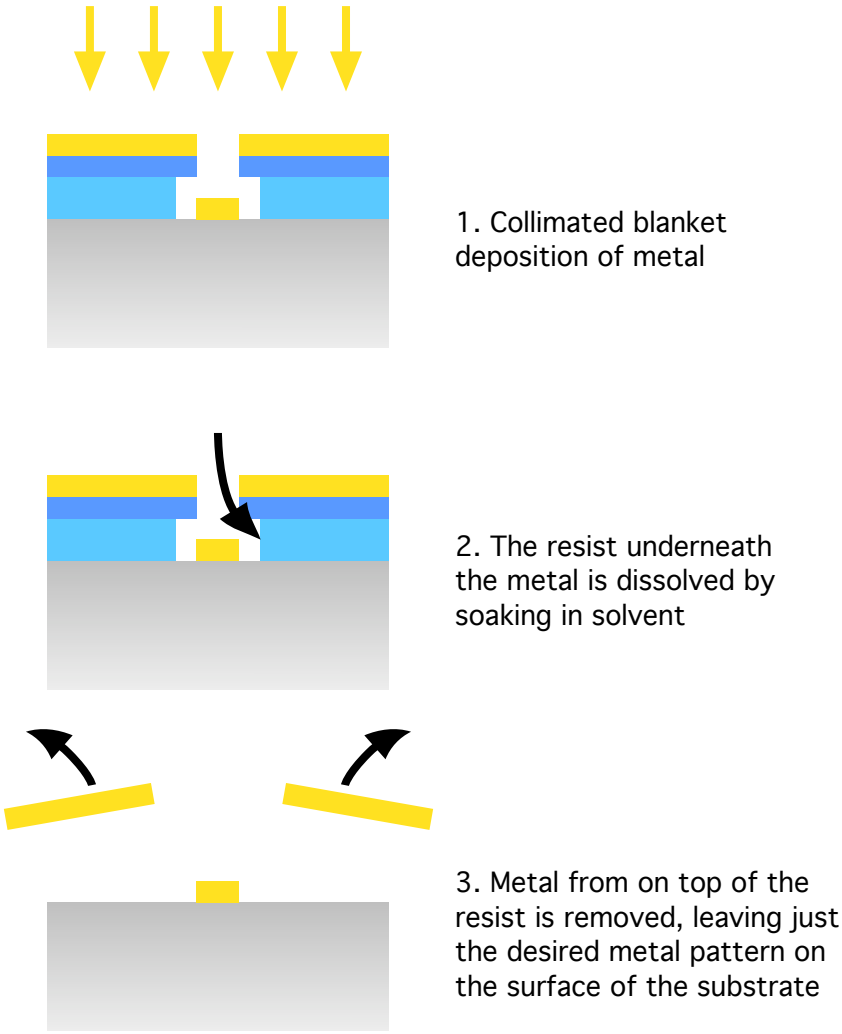


Figure 2.12: Diagrammatic representation of a lift-off process for transferring the pattern from resist into metal.

metal to be deposited. As the crucible and metal heats up the metal begins to evaporate, and travels away from the sample. The chamber is evacuated so that the mean free path of the evaporated metal atoms is greater than the distance separating the crucible from the sample within the machine. In this way the atoms travel in a straight line from the crucible to the sample without scattering off the remaining air molecules and therefore remain well collimated at the surface of the substrate. The rate of deposition and hence of evaporation can be measured using a Quartz crystal microbalance in the path of the evaporated metal. The shear mode acoustic resonance of the microbalance is affected by changes in its mass and hence by monitoring the resonant frequency of the crystal the amount of metal deposited can be measured. Furthermore, the rate of metal being deposited can be controlled by adjusting the power of the electron beam source thereby changing the temperature of the melt, using the crystal as the controlling element in the feedback loop.

Use was also made of a "sputter coater" to deposit metal films. Sputter coaters work by bombarding a solid target of the metal to be deposited with energetic ions formed in a plasma of an inert gas. The energetic ions eject atoms from the metal target which then travel towards the substrate undergoing scattering events in the plasma. The metal atoms arrive at the substrate's surface from a range of trajectories and therefore the layer formed by sputtering is conformal and unsuitable for a lift-off process. However, metals with high boiling points that cannot easily be evaporated can be deposited by sputtering, and compounds such as Al_2O_3 can be deposited by reactive ion sputtering by introducing reactive gases to the plasma [92].

Etching is the second method of pattern transfer and it is a subtractive method. Starting with a blank substrate a blanket deposition of metal is performed. (Although the example of a metal layer is used here it is possible to deposit and etch a wide range of materials from metals, oxides, nitrides and other dielectric materials, through to resists, or even semiconductor substrate material.) The resist is coated onto the

metallised substrate and lithography is then performed. After development the substrate has regions where the underlying metal surface is exposed and regions still coated with resist. The exposed metal regions are then removed in an etchant while the resist protects and prevents the removal of the covered metal regions.

The etchant used can be either a chemical solution, in which case the process is known as "wet-etching", or it could be a gas or plasma etchant, in which case the process is called "dry-etching". In wet-etching, the substrate is submerged in a solution in which the metal is soluble but the resist is insoluble (or at least the rate of dissolution of the resist is appreciably less than that for the metal). The metal exposed to the solution is removed and once the resist is removed metal is left in the inverse pattern of the exposed regions (assuming the use of a positive resist). Dry-etching can work in two ways, chemically or physically, but often the dry-etching process uses a combination of both. Chemical dry etching works in a similar method to wet-etching, in that a reaction occurs between the molecules in the gas and in the metal being etched which forms a volatile product, removing the metal from the surface of the substrate. Physical etching is somewhat different in that it accelerates ionised gas molecules towards the substrate's surface such that when the molecules impact the metal some of the metal is sputtered away from the surface. The resist protects the unexposed regions in both cases and again, once the resist is removed, a copy of the exposed pattern is left in the metal layer.

Both wet and dry-etching can be either "isotropic" or "anisotropic" etch processes. An isotropic etch process means that the etch rate of the material is not dependant on the direction that is being etched. This results in some side-ways etching of the material. With anisotropic etching, on the other hand, the etch rate is dependant on the orientation of the sample with respect to the incident gas molecules or ions or with the crystallographic orientation of the substrate material.

Having introduced the basic tools used in fabrication, some of the more advanced electron beam lithography topics will now be discussed. The alignment process will be

examined, describing how marker are used and detected and some of the limitations of conventional alignment schemes will be discussed, concluding with the introduction of an alternative method of finding markers.

2.2 Alignment

The fabrication of functional semiconductor devices often requires multiple lithographic exposures, each of which is followed by some pattern transfer process that alters the areas of the sample as defined by the exposure. It is critical that the structures defined in these different lithographic steps are positioned accurately with respect to each other. A common example of this would be in the fabrication of a field effect transistor where the drain and source are patterned during the first lithographic exposure and a subsequent exposure is used to define the gate of the transistor which must be accurately positioned between the already defined drain and source regions to produce a working device. This process of positioning a pattern during exposure so that it is matched to some existing features is known as "alignment".

Alignment can correct for several different types of distortion allowing the pattern being exposed to match those already exposed. The simplest distortion encountered is a translational offset, illustrated in the first row of distortions shown in figure 2.13. With a translational offset the pattern to be exposed is misplaced horizontally, vertically or a combination of the two with respect to the previously exposed pattern. Translational shifts can be introduced for a number of reasons. Firstly the mounting of the substrate on a holder introduces an unknown element into the sample's position, secondly thermal expansion of the holder would cause the substrate to be at an unknown position, and lastly any charge in the column, on the apertures or of the substrate could cause a translational shift of the beam with respect to the sample.

The second class of distortions, scale errors, can occur quite easily in electron beam lithography systems. The need for a height meter was discussed in section 2.1.1.6, and

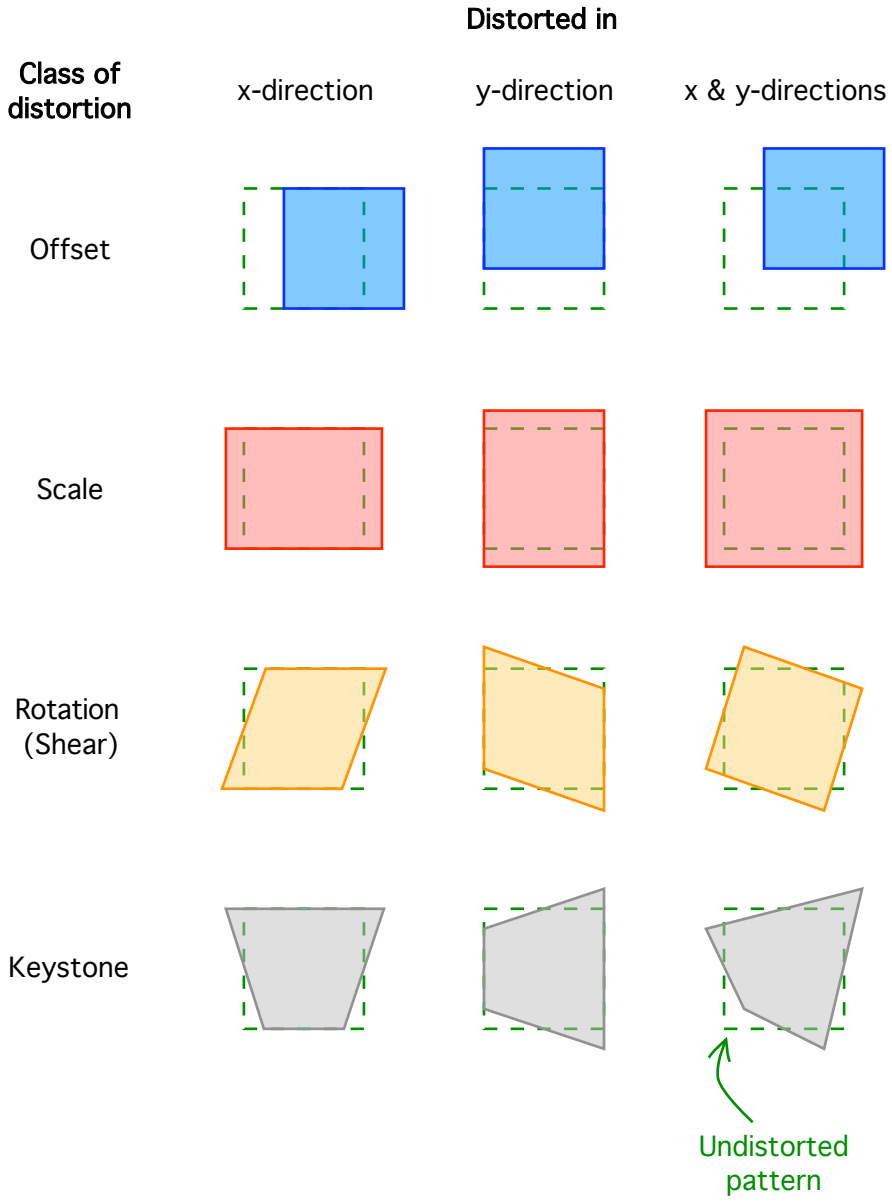


Figure 2.13: Illustration of the possible distortions that can occur to a previously exposed pattern in relation to the pattern to be exposed.

it was noted that variations in the distance between the substrate surface and the final lens caused scale changes in the patterns written on the substrate. Although the height meter takes readings in realtime during an exposure and applies corrections for these errors the laser spot is relatively large and so the correction provided is not perfect. Height variations between the two exposures serves as an example of how scale errors can be introduced. The second row in figure 2.13 shows x, y and combined scale errors for a simple pattern.

Loading misalignments when mounting samples on holders can also introduce rotational distortions, i.e. when a substrate is mounted at a slightly different angle during successive exposures. Although care is taken when the samples are mounted on the holders, there can be rotational misalignments of up to $0.2\,^{\circ}$, the maximum allowed by the VB6 specification. These fixed-body rotational distortions, are a sub-set of a class of distortions called "shear distortions". The fourth row in figure 2.13 shows how individual x and y shears can be combined to form a fixed-body rotation explaining the commonly used term for these distortions.

Lastly there are keystone distortions which occur whenever the substrate surface is tilted with respect to the final lens. The fourth row of figure 2.13 shows the effect of x, y and combined keystone distortions.

There will always be a translational shift due to the way that samples are mounted during exposure and often the mounting procedure introduces a rotational element to the displacement as well. More complex displacement effects, which include shear and keystone effects, can be introduced due to slight deformations of the substrate that stretch and skew the exposed pattern away from the ideal. More complex distortions result from bowed wafers and these introduce barrel and pincushion effects as well. All these effects can reduce the placement accuracy over the entire area of a pattern and if left uncorrected can result in the exposed pattern not matching the intended design.

It is convenient to think of positions on each of the two layers to be aligned as being

described by different co-ordinate systems. The design has an idealised, undistorted co-ordinate grid which is what we expect to observe on the wafer after exposure. However, due to the distortions described above, this idealised co-ordinate system, when measured on the sample's surface is offset, rotated, stretched and skewed and subjected to more complex keystone, barrel and pincushion effects. The process of alignment can then be thought of as finding the transform to convert one co-ordinate set to the other, so that when subsequent layers of the design are exposed, the pattern is written with the same distortions and therefore exactly matches the existing exposed layers.

This is generally achieved by the use of a "projective transformation" to convert the ideal co-ordinates on the design to match the actual positions as measured on the wafer. Other types of transformation could be used, such as "Euclidean" or "affine transformations", however these would provide corrections for only certain classes of distortions between the co-ordinate frames. A linear transform is assumed sufficient since the misalignments and distortions introduced during wafer processing will in general be small enough that they can be considered linear, i.e. the positional misalignment will always be a linear term and any tilt or rotation also results in a linear distortion across the surface of the wafer. Bending of the wafer would cause barrel and pincushion distortions that are not linear in nature, however, their relatively small magnitude, especially when measured over small regions, means that they can be closely approximated by linear terms.

The linear transformations can be divided into three classes of transformation, with each class having slightly different constituent features and being able to produce slightly different distortions. The simplest is the *Euclidean transformation* which essentially allows for changes of position and orientation between co-ordinate frames. It can also produce reflections however this is of little interest in this case. Euclidean transformations cannot produce scale changes or introduce shearing between the co-ordinate frames and therefore preserve the lengths and angles of objects. The next slightly more general

transformation is the *affine transformation* which can produce position and orientation changes and can scale and skew objects but it preserves the parallelism of lines and objects. While the Affine transform accounts for most of the effects likely to be encountered it cannot provide a method to include keystone effects which can easily be introduced by sample tilt. To account for keystone effects the most general class of linear transformations must be used, that is *projective transformations*. This type of transformation does not preserve the position, orientation, length, angle or shape of an object or the parallelism of lines, but can account for all the distortions mentioned above. It derives its name because it describes all the possible distortions when a 2-D image is projected onto a plane, which is a very close analogy for the ebeam process where the pattern is 'projected' onto a substrate.

The general expressions for a projective transform from the (x, y) co-ordinate frame of the design to the (X, Y) co-ordinate frame on the sample are given by equations 2.5 and 2.6, with the relationship between the coefficients and physical distortions as detailed in table 2.1.

$$X = \frac{a + cx + ey}{1 + gx + hy} \tag{2.5}$$

$$Y = \frac{b + dy + fx}{1 + gx + hy} \tag{2.6}$$

Coefficient	Distortion
a, b	X, Y offset
c, d	X, Y scale
e, f	X, Y rotation (shear)
g, h	X, Y keystone

Table 2.1: Details of projective transformation coefficients

An example of how the projective transformation affects a co-ordinate system is

shown in figure 2.14 which shows how the idealised co-ordinate system of the design, figure 2.14(a), and the distorted co-ordinate system of the previously exposed pattern, figure 2.14(b), can be related by a projective transformation.

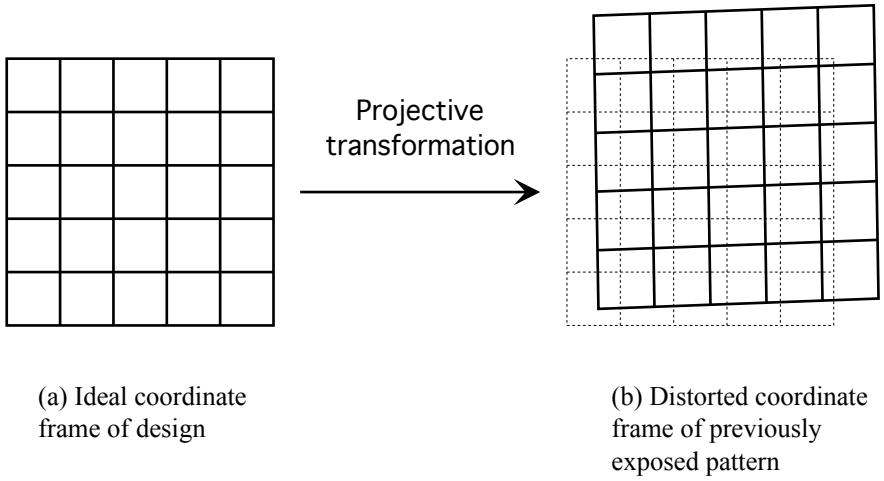


Figure 2.14: Projective transformation relationship between an ideal and a distorted co-ordinate frame.

To be able to apply a projective transformation to convert between two co-ordinate systems requires a method to compute the eight coefficients needed to define the transformation. This requires that eight pieces of information are gathered about the relationship between the two co-ordinate frames. Comparing the locations of four points on the design with the equivalent points on the exposed pattern provides measurements for four relationships between x co-ordinates and four between y co-ordinates. This provides enough information to be able to compute the complete projective transformation between the two frames and hence to be able to align the two layers correctly.

In practice this is achieved by including four markers in the design; usually, one in each of the corners of the pattern. These markers define four reference points on the design and after this first layer has been exposed, the four physically fabricated markers define corresponding reference points on the wafer. The second level of the pattern is designed relative to the same four fixed points as on the first level of the design. These

positions along with the measured positions of the four markers on the wafer are then used to define the mapping transform and hence to align the second level with the respect to the first.

The layout of the markers is important as it defines the sensitivity to each of the distortions discussed above. For instance if the markers are positioned in the centre of the four edges of a square then their positions are not changed by keystone distortions as shown in figure 2.15. While positioning markers further apart increases the sensitivity to scale and rotational distortions it also increases the initial uncertainty of their position. This requires that a larger search area is used, increasing the time required and the area of unwanted exposure. It is also desirable to position the marker close to the region to be patterned so that local variations in the wafer and pattern can be corrected. Positioning the markers in the corners of the area to be patterned is a good compromise, maximising the sensitivity to all classes of distortion while maintaining the speed of detection and the validity of the correction for the region of the sample being exposed.

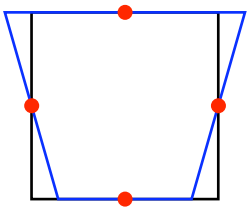


Figure 2.15: Schematic showing that the positions of markers in the centre of the edges of a square are unaffected by keystone distortions.

A relatively straightforward matrix inversion calculation allows the eight coefficients of the projective transformation to be calculated from the eight co-ordinates pairs. The standard Gaussian elimination with back substitution technique [93] is employed to solve this mathematically however in more complex cases there is the possibility of using singular value decomposition for cases where there are fewer or more co-ordinates than coefficients.

Alignment can be achieved with fewer than four markers, however, with a reduced set of information only simpler transformations can be calculated which can only provide corrections for a subset of the distortion effects described above. With four markers, the eight coefficients for offset, scale, rotation and keystone effects can all be calculated and hence a projective transformation can be calculated to correct for all these distortions. Using only three markers means that only six coefficients can be calculated and only three out of the four distortions can be corrected. Reducing the number of markers again, further limits the corrections that can be applied, and in the extreme case of using just a single marker, only two coefficients, usually those of the offsets since this is often the cause of the largest positional errors, can be calculated and only simple translational shifts can be compensated for in the alignment process.

2.2.1 Alignment Markers

The whole of the alignment process as described above relies on the ability to create and then to accurately measure the position of markers on a wafer or sample. Traditionally the markers used are squares or other simple geometric shapes. They are either patterned as a material with a different atomic number or as topographical features, by etching pits into or leaving raised section on top of the substrate. The markers are detected by scanning the same electron beam used to expose the resist in functional parts of the sample across the marker and monitoring the backscattered electron signal as a function of beam position.

The VB6 ebeam lithography column has a four quadrant electron detector that can be used to 'see' the substrate. This is a solid-state semiconductor backscattered electron detector. The detector can be used to turn the electron beam lithography tool into a rudimentary scanning electron microscope, which can detect backscattered electrons and high energy secondary electrons.

When a focussed electron beam is directed at a substrate the size of the point of entry is determined by the size of the focussed beam, however the electrons in the beam interact

with the atoms in the substrate and are scattered over a considerably larger area, on the order of $10\,\mu m^2$ for 100 keV electrons. These interactions can be either elastic, where the incident electron is scattered with no kinetic energy loss, or inelastic, where some of the incident electron's energy is transferred to the atoms in the substrate. Figure 2.16 schematically illustrates the main electron-solid interactions that can occur and shows how some of the electrons emerging from the substrate's surface can be detected.

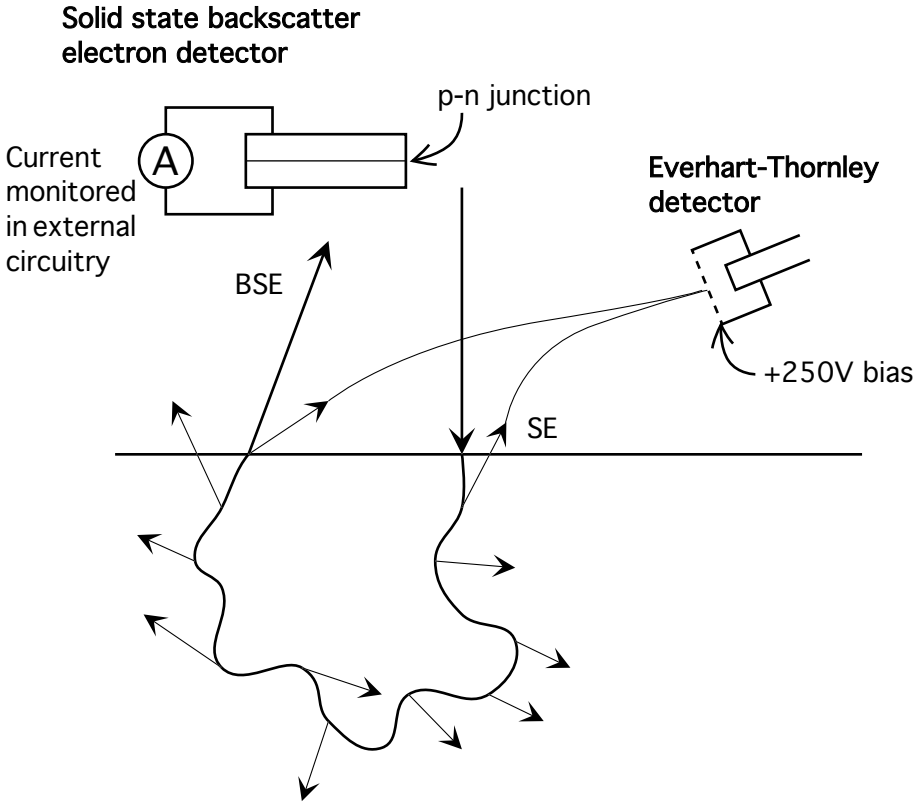


Figure 2.16: Schematic illustration of the main electron-solid interactions showing how secondary electrons and backscattered electrons can be detected.

The elastic scattering events occur when the incident electron passes close to the nucleus of an atom in the substrate and the large electrostatic forces experienced can scatter the incident electrons through large angles. The scattering angle is large enough that when an electron has been elastically scattered up to a few times its trajectory can

have been altered enough that it can escape from the surface of the substrate. Electrons scattered in this way which still have a significant amount of kinetic energy as they escape from the surface back towards the direction of incidence are said to have been "backscattered" and are called "backscattered electrons" (BSE). The probability that an incident electron will be elastically scattered increases strongly with the atomic number of the material, approximately as Z^2, but decreases strongly as the electron energy is increased, approximately as $1/E_{e^-}^2$.

The other electrons which escape the surface of the substrate are known as "secondary electrons" (SE). These electrons are ejected from the substrate as a result of the kinetic energy transferred from a beam electron during an inelastic scattering event. They are commonly used in SEM imaging since their lower energies mean that a moderately biased detector (+250 V) can collect >90% of the secondary electrons generated. The topography of the substrate as well as the local potential or charging affects SE production and so SE images are formed from different contrast mechanisms to BSE images. BSEs have a much higher energy, typically >50% of the incident beam energy and therefore their trajectory is unaffected by moderately biased detectors and only BSEs emitted in line-of-sight of the detector can be collected. Backscattered electrons can be detected by using a scintillator material connected to a photomultiplier tube or, by using a target material with a high SE yield, the BSEs can be converted into multiple SEs which can then be detected, for instance using an Everhart-Thornley detector [94]. Alternatively since they have a large energy they can be directly detected using a solid state semiconductor detector, and this is the detection method used by the electron beam lithography tool in this study. Annular backscatter detectors are fitted below the final lens just above the level of the substrate and so subtend a large solid angle to the substrate surface collecting a large proportion of the generated BSEs emerging from it.

The detector works by harnessing the fact that energetic electrons will generate electron-hole (e-h) pairs within a semiconductor material due to inelastic scattering pro-

moting a valence band electron into the conduction band, leaving behind a hole in the valence band. The energy required to produce such an e-h pair is about 3.6 eV for a silicon detector, and so a single 100 keV BSE has the potential to produce more than 27,000 electrons in the detector. Applying a bias to the detector stops the generated e-h pairs from immediately recombining by attracting the electrons and holes towards opposite sides of the device. This motion of the charged particles is detectable as a current in external circuitry and this current is directly proportional to the number of incident BSE.

2.2.2 Contrast Mechanisms

Electron scattering within a material is governed by Rutherford scattering formula which states that the number of electrons scattered through a given angle is proportional to the square of the atomic number of the material doing the scattering. Hence, by using a material for the markers that has a different atomic number from the surrounding substrate a different number of electrons will be backscattered when the electron beam is positioned on the marker. This manifests itself as a change in the magnitude of the backscattered electron signal detected inside and outside the mark [95]. For a topographical mark, a slightly different phenomenon occurs as described by Wilson et al. [96]. This results in a detected backscattered electron signal which has spikes corresponding to the locations where a step change in the topography exists.

Figure 2.17 shows schematic cross-sections of the two types of markers and illustrates the BSE signal received by the backscattered electron detector when the beam is scanned across each marker. Analysis of this detected signal allows the marker's edges to be found, either by examining the signal levels, or the rate of change of the signal. Once the positions of the marker edges are known, a simple calculation allows the location of the centre of the marker to be established. For reasons of noise reduction the beam is generally scanned across each edge of the marker several times in both the x and the y directions and the edge locations extracted from the average of the whole set of scans.

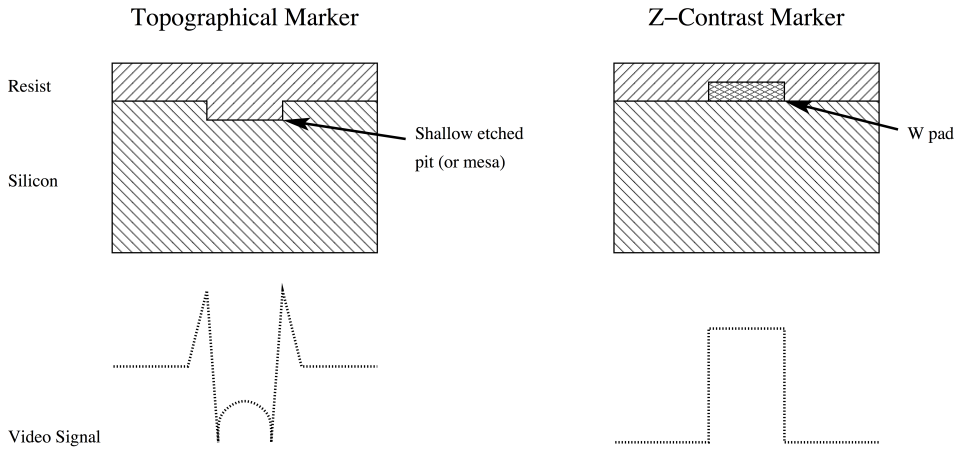


Figure 2.17: Topographical and material markers and their detected backscattered electron signals.

2.2.3 Current Limitations

While the current alignment methods have provided sufficiently accurate alignment for a wide range of devices the accuracy which is ultimately attainable using this process of alignment is inherently limited. One of the fundamental problems is that the marks only contain useful information about their position, and hence the alignment, at their edges, and therefore the alignment accuracy is directly related to how well the patterned edges are defined which is made worse by the need for thick metal layers to provide good contrast. As well as this the number of edges within the marker greatly affects the accuracy with which it can be located since more edges effectively increases the signal to noise ratio. The maximum number of edges within a given interrogation area would be formed by a 1:1 mark:space ratio grating however the potential problem with a grating is that it is a periodic structure and unless the entire grating area is interrogated the found marker position could be offset by an integer number of periods. There also needs to be a relatively fast method of retrieving the extra information that multiple edges provide, as longer or repeated line scans will become prohibitively time-consuming not only to collect but also to analyse.

Furthermore, to find a marker's position the small region which contains all the positional information at the edges of the mark must be heavily interrogated and this often requires use of a large beam current to provide sufficient signal to noise ratio. The interrogation of the edge frequently leads to the exposure of this area of resist and hence the marker becomes exposed to whatever the subsequent process step requires. This, almost inevitably, results in the edges of the marker being destroyed or damaged and while the marker may continue to be usable the accuracy with which it can be located will be degraded. Which leads to another fundamental inaccuracy in the current alignment method: when multi-stage alignments are required, different alignment marks must be used for each alignment and this introduces a further source of errors into the process.

To overcome these limitations Holburn et al. [13] proposed the use of techniques from the image registration and pattern recognition fields, by suggesting that an alignment algorithm based on the process of correlation could be used to perform precise alignment in electron beam lithography.

2.2.4 Correlation

Correlation, in image processing terms, is a process that measures the similarity between two images. As such it can be used to locate a marker by correlating a reference image of the marker within an image of the marker as exposed on the wafer with the associated distortions present. Fundamentally this problem of finding a marker at an unknown position in a noisy image is remarkable similar to the use of a linear filter to detect if a known signal is present in a received waveform in the midst of noise and with an unknown delay. This situation has been widely discussed [42, 97–100] particularly in the radar detection context, where it is referred to as the "simple binary detection problem". This essentially asks if a detected waveform contains a known signal plus noise or just noise, and locates the position of the signal if present. This is precisely the task required to locate a marker in an image, albeit in 2-dimensions rather than one.

2.2.4.1 Matched filters

Consider the case of figure 2.18 where a received waveform, $x(t)$, consists either solely of white Gaussian noise, $n(t)$, or of $n(t)$ plus a signal, $s(t)$, of known form. Using a linear filter it is desired to be able to detect when the signal is present in the received waveform. Turin [97] shows that the filter that maximises the signal-to-noise ratio of the output, $y(t)$, is the matched filter.

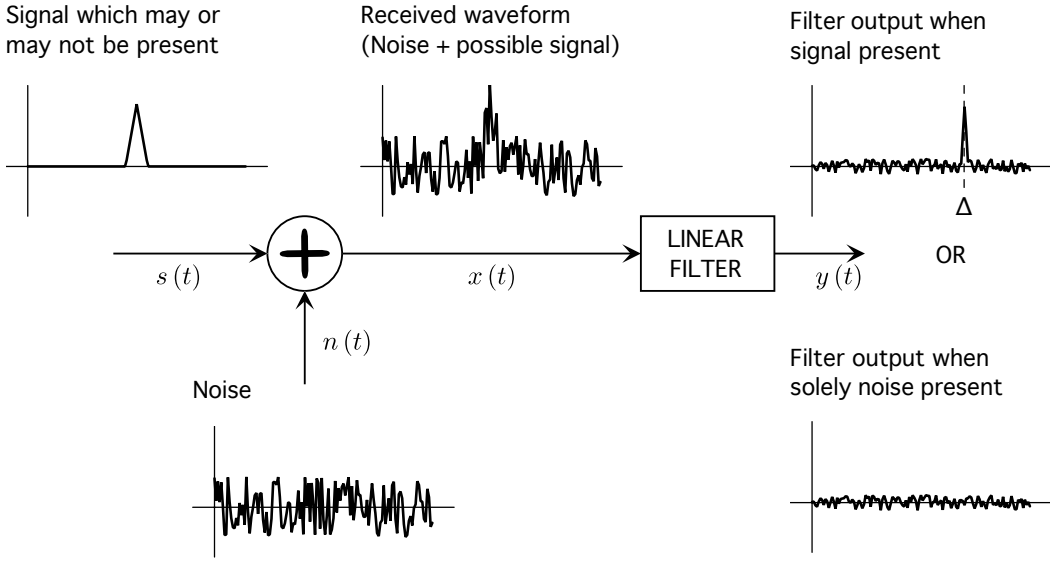


Figure 2.18: Use of a linear filter to detect the presence of a signal of known form in a noisy received waveform.

Given a function, $s\left(\hat{\mathbf{t}}\right)$, then its matched filter is defined as having an impulse response, $h\left(\hat{\mathbf{t}}\right)$, given by equation 2.7,

$$h\left(\hat{\mathbf{t}}\right) = ks\left(\Delta - \hat{\mathbf{t}}\right), \tag{2.7}$$

where k and Δ are arbitary constants. Examining this in the context of our detection system then the transfer function of the matched filter, which is the Fourier transform of the impulse response, has the form

$$
\begin{aligned}
H\left(j2\pi\hat{\mathbf{f}}\right) &= \int_{-\infty}^{\infty} h\left(\hat{\mathbf{t}}\right) e^{-j2\pi\hat{\mathbf{f}}\hat{\mathbf{t}}} \mathrm{d}\hat{\mathbf{t}} \\
&= k\int_{-\infty}^{\infty} s\left(\Delta - \hat{\mathbf{t}}\right) e^{-j2\pi\hat{\mathbf{f}}\hat{\mathbf{t}}} \mathrm{d}\hat{\mathbf{t}} \\
&= ke^{-j2\pi\hat{\mathbf{f}}\Delta}\int_{-\infty}^{\infty} s\left(\hat{\mathbf{t}}'\right) e^{-j2\pi\hat{\mathbf{f}}\hat{\mathbf{t}}'} \mathrm{d}\hat{\mathbf{t}}', \qquad (2.8)
\end{aligned}
$$

where the substitution $\hat{\mathbf{t}}' = \Delta - \hat{\mathbf{t}}$ has been made. The Fourier transform of the original function is

$$
S\left(j2\pi\hat{\mathbf{f}}\right) = \int_{-\infty}^{\infty} s\left(\hat{\boldsymbol{t}}\right) e^{-j2\pi\hat{\mathbf{f}}\hat{\mathbf{t}}} \mathrm{d}\hat{\boldsymbol{t}}. \qquad (2.9)
$$

Comparing this with equation 2.8 shows that

$$
\begin{aligned}
H\left(j2\pi\hat{\mathbf{f}}\right) &= ke^{-j2\pi\hat{\mathbf{f}}\Delta} S\left(-j2\pi\hat{\mathbf{f}}\right) \\
&= ke^{-j2\pi\hat{\mathbf{f}}\Delta} S^{*}\left(j2\pi\hat{\mathbf{f}}\right), \qquad (2.10)
\end{aligned}
$$

namely, that except for an amplitude and delay factor the transfer function of a matched filter is the complex conjugate of the Fourier transform of the signal it is matched to. Hence matched filters are often referred to as "conjugate" filters.

2.2.4.2 Properties of Correlation

Mathematically, correlation is closely related to convolution as can be seen from equations 2.11 and 2.12, the convolution and correlation integrals respectively for two functions $f\left(\hat{\mathbf{r}}\right)$ and $g\left(\hat{\mathbf{r}}\right)$, where '$*$' denotes convolution and '$\star$' denotes correlation [101].

$$f(\hat{\mathbf{r}}) * g(\hat{\mathbf{r}}) = \int f(\hat{\mathbf{s}})\, g(\hat{\mathbf{r}} - \hat{\mathbf{s}})\, \mathrm{d}\hat{\mathbf{s}} \tag{2.11}$$

$$f(\hat{\mathbf{r}}) \star g(\hat{\mathbf{r}}) = \int f(\hat{\mathbf{s}})\, g(\hat{\mathbf{r}} + \hat{\mathbf{s}})\, \mathrm{d}\hat{\mathbf{s}} \tag{2.12}$$

Convolution is often expressed in a more convenient form using the Fourier transforms of the functions, as in equation 2.13, where $F\left(\hat{\mathbf{k}}\right)$ and $G\left(\hat{\mathbf{k}}\right)$ denote the Fourier transforms of $f(\hat{\mathbf{r}})$ and $g(\hat{\mathbf{r}})$ respectively, and $\mathcal{F}\{\ \}$ denotes a Fourier transform. Using the convolution theorem we can write equation 2.13, which states that the convolution of two functions in the spatial domain is equal to the multiplication of their Fourier transforms in reciprocal space.

$$f(\hat{\mathbf{r}}) * g(\hat{\mathbf{r}}) = \mathcal{F}\left\{ F\left(\hat{\mathbf{k}}\right) G\left(\hat{\mathbf{k}}\right) \right\} \tag{2.13}$$

A similar relationship exists for correlation, as shown in equation 2.14. The sign change in the correlation integral, equation 2.12, manifests itself as a complex conjugation, denoted by the superscript asterisk, of one of the functions in the Fourier domain.

$$f(\hat{\mathbf{r}}) \star g(\hat{\mathbf{r}}) = \mathcal{F}\left\{ F\left(\hat{\mathbf{k}}\right) G^{*}\left(\hat{\mathbf{k}}\right) \right\} \tag{2.14}$$

When $g(\hat{\mathbf{r}})$ is set as the pattern to be detected and $f(\hat{\mathbf{r}})$ is an image of the pattern with added random noise, equation 2.14 shows that the correlation is mathematically equivalent to the result that would be obtained by matched filtering. Hence correlation can be considered as acting as a matched filter with the same optimum signal-to-noise properties.

It is sometimes useful to investigate the "autocorrelation" of a function, which is the correlation of a function with itself, the reason being that the autocorrelation has an important relationship to the power spectral density of a function. This can be most

easily demonstrated if we consider how equation 2.14 changes in the case of correlating two equal functions.

$$\begin{aligned} f(\hat{\mathbf{r}}) \star f(\hat{\mathbf{r}}) &= \mathcal{F}\left\{F\left(\hat{\mathbf{k}}\right) F^{*}\left(\hat{\mathbf{k}}\right)\right\} \\ &= \mathcal{F}\left\{\left|F\left(\hat{\mathbf{k}}\right)\right|^{2}\right\} \end{aligned} \tag{2.15}$$

Therefore, the autocorrelation is simply the Fourier transform of the power spectral density of the original function. This remarkable result is known as the Wiener-Khintchine theorem. Furthermore the autocorrelation has its maximum at the origin, which can be shown by considering the Cauchy-Schwarz inequality. Equation 2.16 shows the Cauchy-Schwarz inequality after squaring both sides.

$$\left(\int_{-\infty}^{\infty} f(u)\, g(u)\, \mathrm{d}u\right)^{2} \leq \int_{-\infty}^{\infty} f(u)^{2}\, \mathrm{d}u \int_{-\infty}^{\infty} g(u)^{2}\, \mathrm{d}u \tag{2.16}$$

Substituting $f(u+x)$ for $g(u)$ allows this to be rewritten as

$$\left(\int_{-\infty}^{\infty} f(u)\, f(u+x)\, \mathrm{d}u\right)^{2} \leq \int_{-\infty}^{\infty} f(u)^{2}\, \mathrm{d}u \int_{-\infty}^{\infty} f(u+x)^{2}\, \mathrm{d}u\,. \tag{2.17}$$

Now the left-hand side of equation 2.17 is simply the square of the autocorrelation of $f(u)$ evaluated at x, so can be rewritten as

$$(f \star f)^{2}(x) \leq \int_{-\infty}^{\infty} f(u)^{2}\, \mathrm{d}u \int_{-\infty}^{\infty} f(u+x)^{2}\, \mathrm{d}u\,. \tag{2.18}$$

However, for any x,

$$\int_{-\infty}^{\infty} f(u+x)^{2}\, \mathrm{d}u = \int_{-\infty}^{\infty} f(u)^{2}\, \mathrm{d}u\,. \tag{2.19}$$

As well as this,

$$\int_{-\infty}^{\infty} f(u)^2 \, \mathrm{d}u = \int_{-\infty}^{\infty} f(u) f(u) \, \mathrm{d}u$$
$$= (f \star f)(0) \ . \tag{2.20}$$

Substitution equations 2.19 & 2.20 into 2.18 gives

$$(f \star f)^2 (x) \leq (f \star f)^2 (0) \ , \tag{2.21}$$

and since $(f \star f)(x) \geq 0$ we conclude that

$$(f \star f)(x) \leq (f \star f)(0) \ , \tag{2.22}$$

and therefore that the autocorrelation has its maximum at the origin. It follows that when one of the functions is shifted by (x, y) then the maximum of the cross-correlation, $f(\hat{\mathbf{r}}) \star f\left(\hat{\mathbf{r}} + (x, y)\right)$, is at (x, y).

2.2.4.3 Graphical Interpretation

It is interesting to consider the process of correlation graphically. Correlation, in essence, provides a measure of similarity between two functions, or images. Each point in the correlation give a measure of the similarity of the two images with a different relative offset, therefore the origin in the correlation measures the similarity when neither image is displaced and a point (a, b) in the correlation measures the similarity when there is a relative displacement of (a, b) between the two images. As such finding the peak in the correlation allows the relative offset between the two images which gives the greatest degree of similarity to be obtained. Therefore, it can be used to locate a reference pattern within another image, which may be noisy or imperfect in other ways. Schematically this is illustrated in figure 2.19, which shows how a simple square reference pattern can be located within a larger image, by finding the point with the greatest brightness in the

correlation.

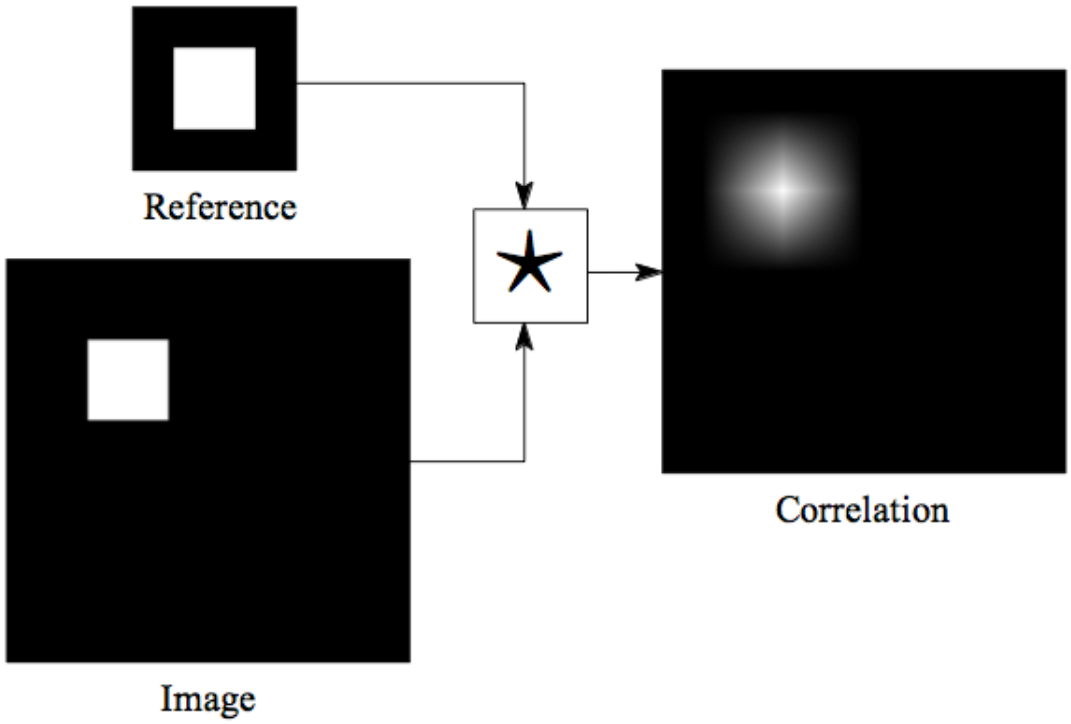


Figure 2.19: Schematic illustration of image correlation.

Correlation is not limited to using simple geometric shapes as reference patterns and it is well known that the pattern used as a reference has a critical influence on the performance of the correlation [14, 17]. Correlation based methods have been used for alignment in electron beam lithography, most recently by Anderson et al. [18] and previous work has noted that the use of more sophisticated marker patterns leads to an improvement in the alignment accuracy [16]. As well as this, since correlation uses information from a much larger area, the careful choice of marker can be used to boost the signal to noise ratio [15]. This may allow the use of lower interrogation doses and introduces the possibility of repeatedly using the same marker, without exposing it sufficiently to modify the marker, in multi-stage alignment, again improving the accuracy.

Perhaps, the most important factor determining the performance of correlation based

alignment is the choice of the pattern used as a reference. However to date, there has been no detailed study into how the design of a reference pattern affects the attainable accuracy of correlation or of an alignment process. The following chapter deals with this topic in depth, examining various families of patterns, considering their correlation properties and their suitability for use as electron beam alignment markers.

3 Marker Patterns for Alignment

A surprising number of patterns have been used as alignment markers. These range from simple geometric shapes, such as squares and octagons discussed in section 3.1.1, to complex alignment targets that have been specifically designed to give as much positional information as possible, such as the 'Universal Alignment Target' discussed in section 3.1.6. There are also several situations where a correlation process or similar is applied to identify a signal or to allow improvements to be made to the performance of a system, in particular matched and correlation filters in communications systems and "Coded Aperture Imaging" in X-ray astronomy. These seemingly unrelated fields have produced some interesting patterns that have desirable features which could be exploited by an alignment process based on correlation.

3.1 Marker Patterns

In this section various classes of pattern will be introduced. The origin of each class of pattern will be discussed and the properties which make it desirable for a correlation based alignment process will be explained. As well as this, examples of typical patterns from each class will be displayed.

3.1.1 Simple Geometric Shapes

Perhaps the simplest but also the most widely used class of marker patterns are the simple geometric shapes. This class includes objects such as squares, octagons, and single or multi-limbed crosses, examples of which can be seen in figure 3.1

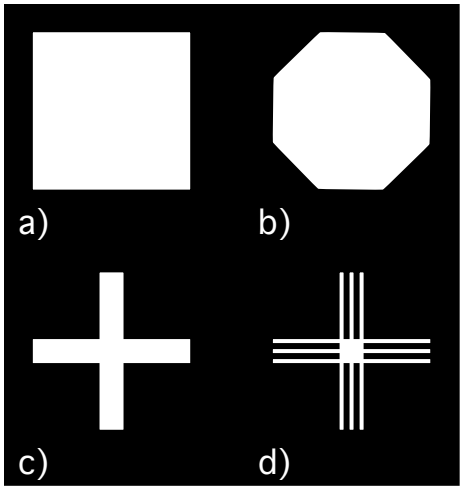


Figure 3.1: Simple geometric shapes that have been used as alignment markers. a) square, b) octagon, c) simple cross, d) multi-limb cross, where each limb is comprised of three lines.

This type of alignment marker has been widely used in e-beam lithography, principally due to the relative ease with which their position can be determined. Using the square marker as an example, figure 3.2 shows the detected BSE signal when the electron beam is scanned across the marker. The positions of the rising and falling edges equate to the marker edges and these positions can simply be averaged to give a measure of the x-position of the centre of the marker. A similar scan in the vertical direction allows the y-position to be found.

More detailed scans can be performed to locate the edges to greater precision and multiple scans in different positions can be used, with a knowledge of the marker geometry, to improve the immunity to noise and defects. The more edges that the marker pattern contains then the more measures of the marker position can be made, so the square has four edges, two in x and two in y. The multi-limb cross of figure 3.1 has twenty-four edges (not counting the ends of the limbs since these are not used during the alignment), twelve each in x and y, therefore contains much more positional information.

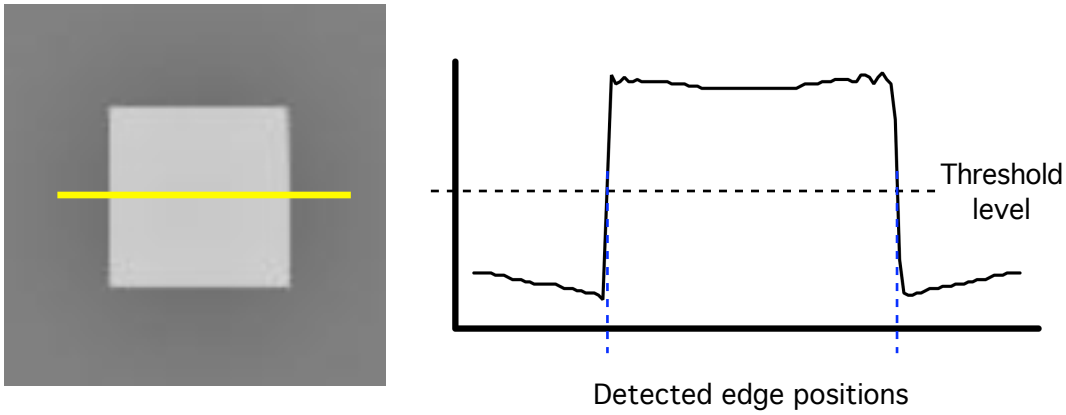


Figure 3.2: Detected backscattered electron signal when the electron beam is scanned across a square marker.

3.1.2 Non-redundant and Uniformly Redundant Arrays

In high-energy astronomy pinhole cameras are often used because they don't rely on refraction and can therefore be used to form images from X-ray and gamma ray sources [102–104]. However the problem with pinholes is that they are small and hence let very little radiation through to the detector. Making the pinholes larger results in a decrease in resolving power, so the bigger the pinholes the more blurred the image becomes. There is a solution, known as "Coded Aperture Imaging" (CAI), which allows the good signal-to-noise ratio of a large collection area to be combined with the superior spatial resolution of small apertures. It works on the principle illustrated in figure 3.3, where an image of the source is formed on the screen by an array of small apertures.

The image formed appears to be an incoherent montage of overlapping images however if the pattern of the pinholes is carefully chosen it is possible to reconstruct a single image with a resolution equal to that of a single pinhole but with an intensity corresponding to the number of pinholes used. The image, I, formed of an object, O, using an aperture pattern, A, can be described as,

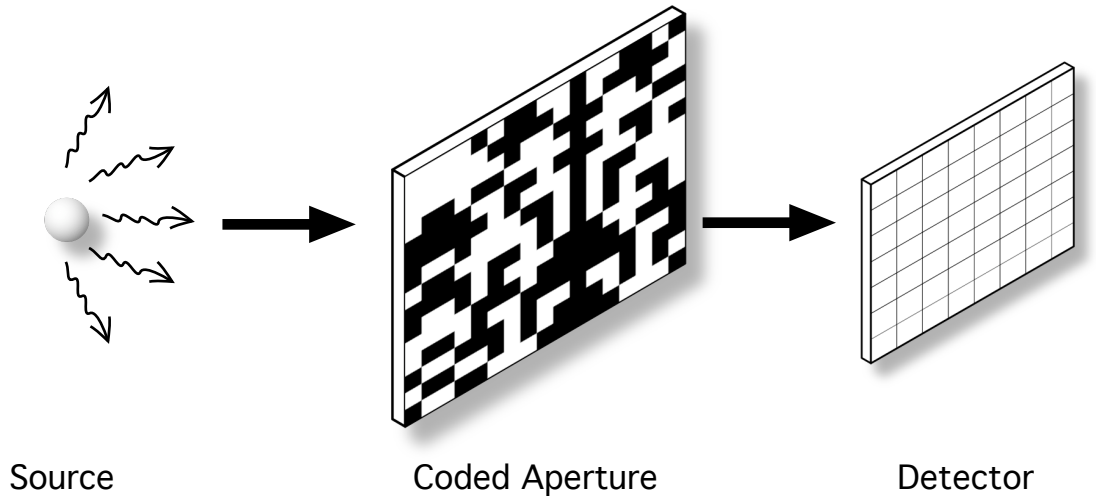


Figure 3.3: Schematic of the principle of operation of Coded Aperture Imaging.

$$I = (O \star A) + N, \tag{3.1}$$

where $\star$ is the correlation operator and N is a noise function. Both deconvolution and correlation can be used to reconstruct the object however only the correlation method will be discussed due to its simplicity. For the correlation method equation 3.1 is correlated by a recovery function, G, giving an expression for the reconstructed object, $\tilde{O}$, as follows,

$$\tilde{O} = I \star G = O \star (A \star G) + N \star G. \tag{3.2}$$

In the ideal case $A \star G$ would be a delta function and the expression for the reconstructed object becomes,

$$\tilde{O} = O + N \star G, \tag{3.3}$$

i.e. a perfect reconstruction of the original object except for the noise term.

The recovery function used must be derived from the aperture pattern, and so the

pattern must be chosen such that $A \star G$ is effectively a delta function. Non-redundant arrays (NRA) have the property that their autocorrelation consists of a sharp central spike with sidelobes equal to unity out to a particular spatial frequency, and then equal to zero or unity beyond that. This is a good approximation for a delta function with an easily removable dc offset. Although the non-uniform sidelobes do introduce some inherent noise, if the pattern is made large with respect to the object this effect becomes negligible.

The autocorrelation properties stem from the fact that if the separations between all possible pairs of holes in an NRA are measured then each separation (up to a given value) will occur once and only once within the pattern, thus the separations are non-redundant. This provides the major advantage to using an NRA however it also enormously restricts the number of holes within a given area as figure 3.4 a) shows.

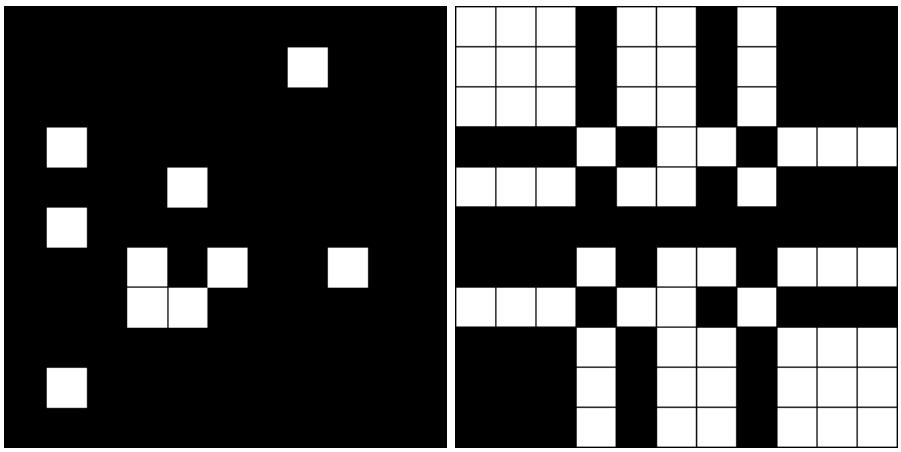


Figure 3.4: Examples of a) a non-redundant array with 10 pinholes, and b) a uniformly redundant array of approximately the same area as a) but with an array of 11x11 pinholes and 50% fill.

An extension to NRA are uniformly redundant arrays (URA) which instead of a single instance of each separation they have the same number of each separation present in the array. In a similar way to the NRA the autocorrelation of a URA is also sharply peaked in the centre, however its sidelobes are not equal to unity, they are uniform but

at a level corresponding to the number of pairs of holes with duplicate separations. This allows them to maintain the δ-function like property but maximise the light collecting area. There are various derivatives of URA, including modified uniformly redundant arrays (MURA) and hexagonal uniformly redundant arrays (HURA) [105–107] which allow more convenient methods of generation to be used or allow a greater density of pinholes and thus physically smaller masks to be used. Several γ-ray and X-ray satellites and balloon-borne telescopes have successfully employed CAI to produce remarkable images of the sky [108].

3.1.3 Pseudo-random Noise Arrays

Pseudo-random noise sequences (PN) have been known to have many useful properties, and have been widely used in communications systems as error checking and correcting codes, cyclic redundancy checks and in spread spectrum systems. These systems all rely on the fact that the autocorrelation of PN sequences is large when there is no shift but uniform and small for any other shifts, i.e. the autocorrelation has a single peak. Perhaps the easiest method to generate PN sequences is through the use of a linear shift-register feedback loop [109], as long as the sequence generated is of maximal length then it will exhibit the good autocorrelation properties above. The PN sequences, as generated, are one-dimensional but there exists a simple method of reforming the sequence into a two-dimensional array with similarly good autocorrelation properties, a pseudo-random noise array [36]. An example PN array is shown in figure 3.5.

In fact PN arrays generated in this way are a sub-class of URA mentioned above in section 3.1.2.

3.1.4 Fractal Patterns

Fractals are patterns which have a similar form at all length scales. That is to say that any small section of a fractal pattern when magnified has the same structure as the original pattern. A good example of this is the Sierpinski carpet fractal, named after the

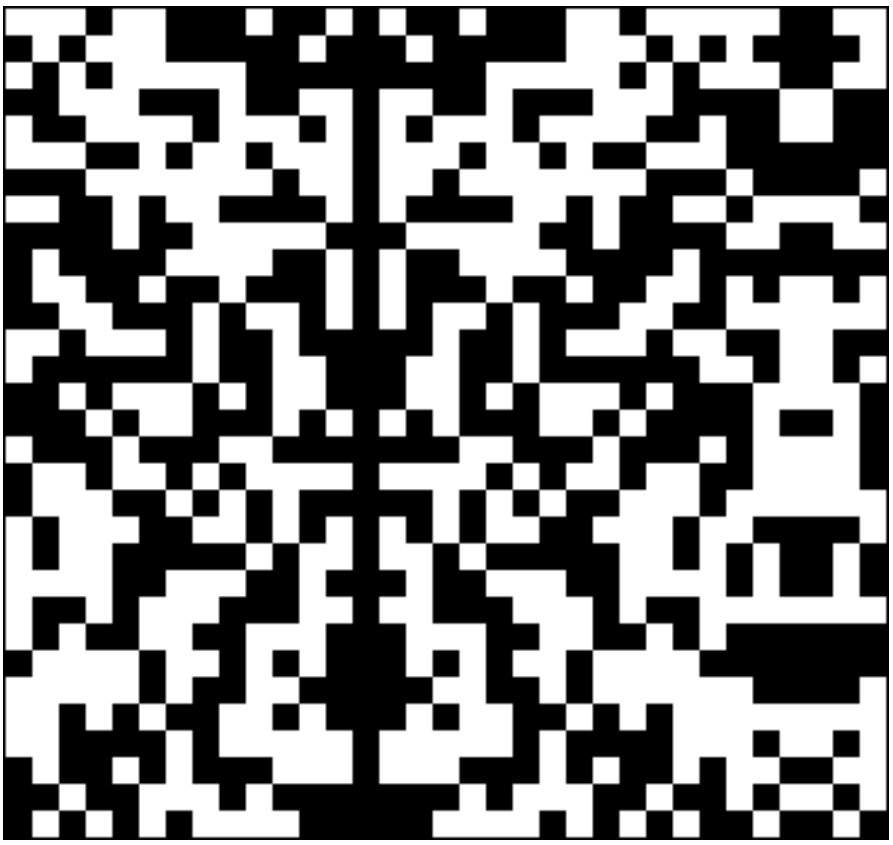


Figure 3.5: Example of a pseudo-random noise sequence formed into an array.

Polish mathematician Wacław Sierpiński, which is shown in figure 3.6. Examining the region outlined in red it is clear that it contains a reduced scale version of the complete pattern, showing the fractal nature of the pattern.

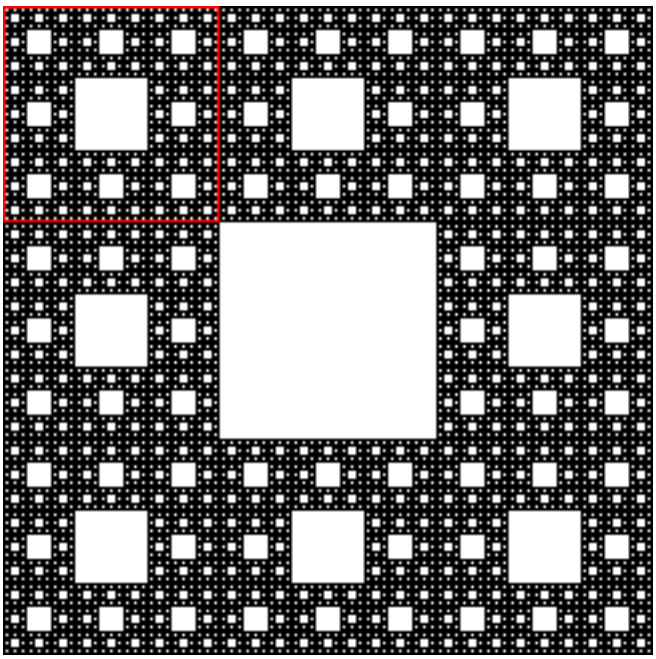


Figure 3.6: A Sierpinski carpet fractal pattern, with a section highlighted to show the pattern's self-similarity.

Fractals have several desirable properties as marker patterns, not least of which is the self-similarity at different length scales. This would mean that exactly the same performance is obtained no matter what magnification is used. This is also the greatest weakness, since if too small a region is interrogated it is impossible to tell uniquely which part of the pattern it comes from, therefore a large fraction of the pattern must be interrogated so that at least one of each size of feature is detected. It is also difficult to optimise the fabrication of fractal patterns since they have both very large and very small elements, and so are limited by both the resolution of the fabrication process and the proximity effect from having large features near to smaller ones.

The centre of mass of a fractal pattern such as the Sierpinski carpet, is at the centre

of the pattern and so this can be used to roughly locate the pattern's centre. The large degree of redundancy in the information content in fractal patterns gives them a good degree of tolerance to missing parts of the pattern, since if some of the smaller parts are obscured or missing then there are still larger elements which can be used to locate the marker.

3.1.5 Barker Sequence

A Barker sequence [110] is a finite sequence of integers, each ± 1, such that the "off-peak autocorrelation" of the sequence has a magnitude of less than 1. Given the sequence, $a_0, a_1, \ldots, a_{n-1}$, the kth aperiodic autocorrelation of this sequence is defined as

$$c_k = \sum_{i=0}^{n-1-k} a_i a_j, \quad \text{where } j = \begin{cases} i+k+n & i+k<0 \\ i+k & 0 \leq i+k < n-1 \\ i+k-n & n-1 \leq i+k \end{cases} \tag{3.4}$$

where k is an integer and $|k| < n$. The value when $k = 0$ is therefore n, and is called the "peak" autocorrelation, however for all other values of k, c_k is denoted as the "off-peak" autocorrelations. Any sequence where $|c_k| \leq 1$ for $k \neq 0$ is known as a Barker sequence.

These sequences are specifically chosen because of their good correlation properties, but there is no easy way to generate Barker sequences. There are also just eight known Barker sequences as shown in table 3.1, where + and - are used to represent $+1$ and -1 respectively.

To form a binary image with similar properties to a Barker series we can replace the $+1$ elements with bright regions and the -1 elements with dark regions. Furthermore, Barker sequences are 1-dimensional but can be used to find a position in 2-dimensions by using a pattern composed of two Barker sequences arranged such that there is one sequence in the x-direction and one in the y-direction, as shown in figure 3.7.

n	Barker sequence
2	++
3	++−
4	+++−
4	++−+
5	+++−+
7	+++−−+−
11	+++−−−+−−+−
13	+++++−−++−+−+

Table 3.1: Barker sequences

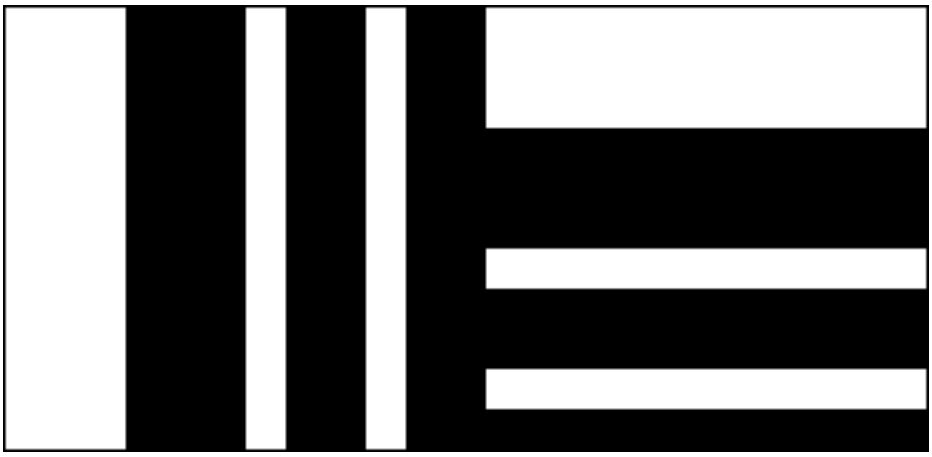


Figure 3.7: A pattern formed from two Barker sequences one in the x-direction and the other in the y-direction.

Barker sequence based markers have already been used as alignment markers in e-beam lithography and have been shown to provide a good level of accuracy. They are limited in that they are 1-dimensional, and therefore multiple Barker sequence markers are needed to provide both x and y positional information, increasing the area that must be interrogated. They are also severely affected by grid degeneracy, see section 3.2.5, which restricts the sub-pixel information that can be obtained from a sampled Barker sequence marker image.

3.1.6 Universal Alignment Target

There has been a substantial amount of work into what alignment target is best to use in machine vision systems. The requirements for a machine vision system are similar to that for an electron beam system however, machine vision systems are in general less restricted. For instance machine vision systems have to cope with alignment that is not uniquely translational, and have to be able to find a marker that could be significantly rotated, skewed by perspective, obscured by other objects, and perhaps of unknown size.

In particular Silver [111] has designed a target which fulfils his criteria of

- high speed of analysis
- high accuracy in all degrees of freedom (translation, rotation, and scaling)
- high success rate (low false detection)
- easy to use in terms of creation, training the machine vision system
- has a well-defined reference point.

The target created is dubbed a “universal alignment target” or UAT.

The UAT designed by Silver is shown in figure 3.8. It has been designed such that it has good sensitivity to rotational, translational, and size changes and that it has good immunity to sampling problems. Further it has few small features making it quick to interrogate but also easy to produce and robust against manufacturing processes.

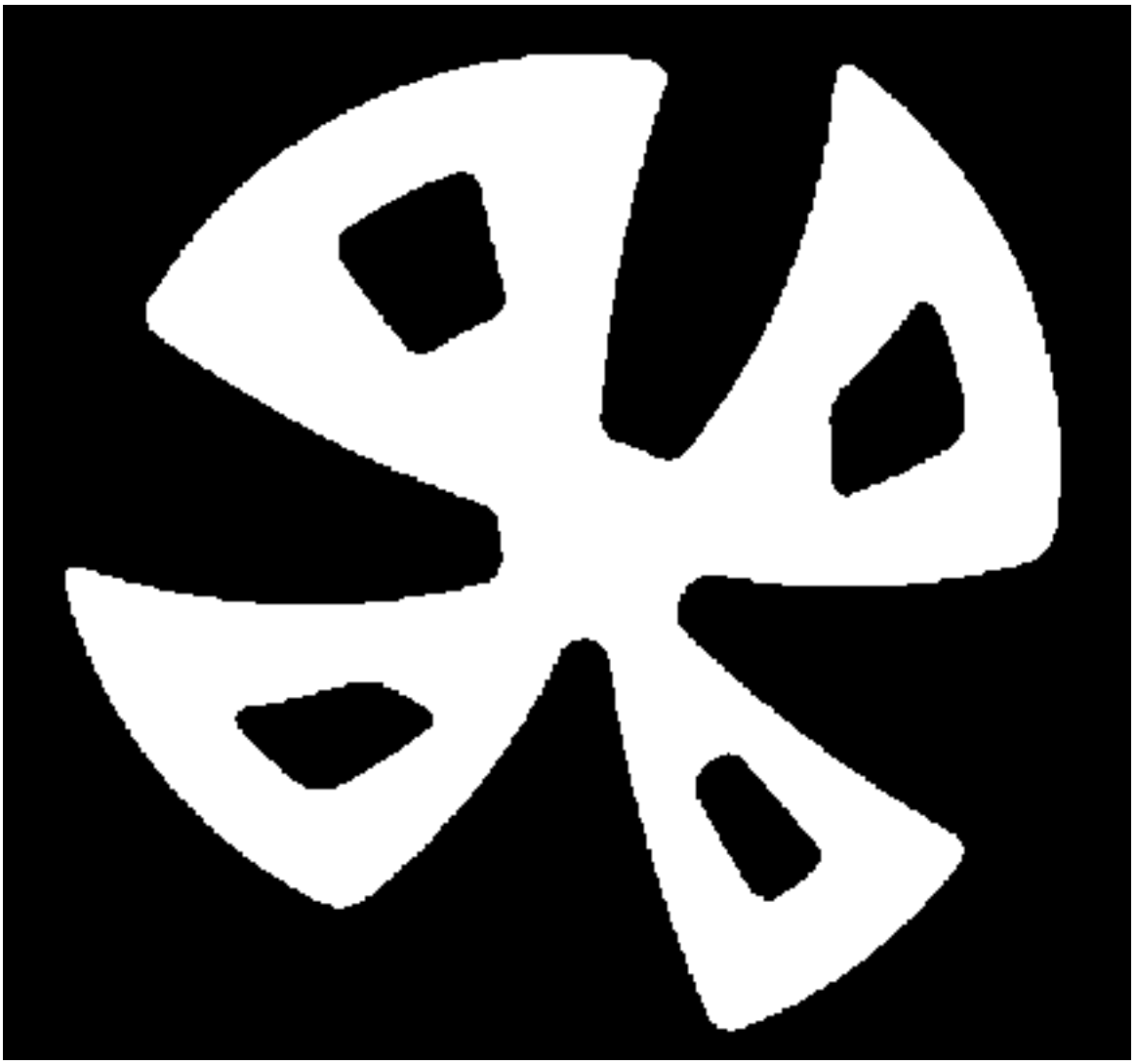


Figure 3.8: Universal alignment target proposed by Silver.

3.1.7 Penrose Patterns

As will be justified in section 3.3, Penrose pattern based markers have proven to be the most suitable, of all the patterns examined, for use as marker patterns in correlation based alignment. Therefore an extensive introduction to Penrose patterns is given here.

A Penrose pattern is derived from a tiling invented by Roger Penrose [112]. The tiling is formed from only two fundamental tiles with simple rules constraining the placement of those tiles such that they can be formed into an infinite tiling of the 2D plane which is translationally aperiodic. This means that every point in the pattern is surrounded by a unique set of tiles and although there is rotational five-fold symmetry in the pattern, no translational displacement of the point of interest will produce another point surrounded by a matching set of the tiles.

One set of fundamental tiles, commonly known as "kites" and "darts", that can be used to form Penrose tilings are shown in figure 3.9, where ϕ is the Golden ratio ≈ 1.618. There is another equivalent set known as "fat" and "thin" rhombi, and the sets can be used interchangeably. The rest of the discussion will use kites and darts with no loss of generality since any pattern made up of one set of tiles can trivially be converted into the other set of tiles. The rules that constrain the placement of tiles is illustrated by the coloured arcs on the tiles in figure 3.9.

Two tiles can only be positioned adjacently if the coloured arcs match across the shared edges. These constraints mean that there are just seven configurations of kites and darts that can be formed around a point; all of which can be used as a starting point, or 'axiom', for an infinite tiling.

The two most commonly used are the "Sun" and "Star" arrangements which are the only two which have five-fold rotational symmetry and a well-defined centre position. They can be used to produce an infinite tiling by a process known as deflation. Deflation takes each of the fundamental tiles in a pattern in turn, and replaces that tile with multiple parts of smaller tiles maintaining the same outline. So a kite is converted into

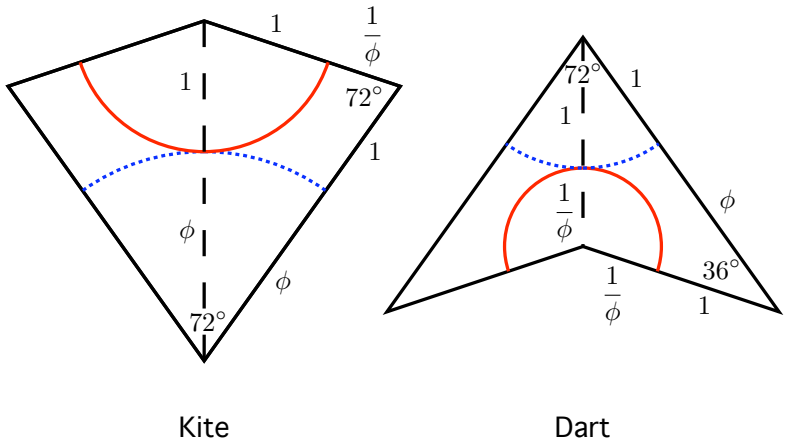


Figure 3.9: The two fundamental tiles that can be used to form Penrose tilings

two kites and two half-darts, while a dart is converted into a kite and two half-darts. This process can be continued infinitely until a pattern has enough detail to cover a given area with a given feature size. The first four generations of patterns from the 'Sun' and 'Star' axioms are shown in figure 3.10.

A Penrose tiling can be used to form a marker pattern by selecting either kites or darts to be metal with the other tile being left blank. An appropriate generation of pattern is chosen so that the desired marker area is covered with features of a given size.

The translational aperiodicity of Penrose patterns gives them an obvious advantage when used as marker patterns since this aperiodicity means that any part of a Penrose pattern is unique. The size of the section examined determines the radius of uniqueness, with larger sections being unique over larger distances. Therefore, with a given extent of pattern there is a minimum area that must be interrogated such that the correlation produces a unique position. This minimum area of interrogation is illustrated in figure 3.11, for a seventh generation Penrose pattern based on the sun axiom with dart elements highlighted.

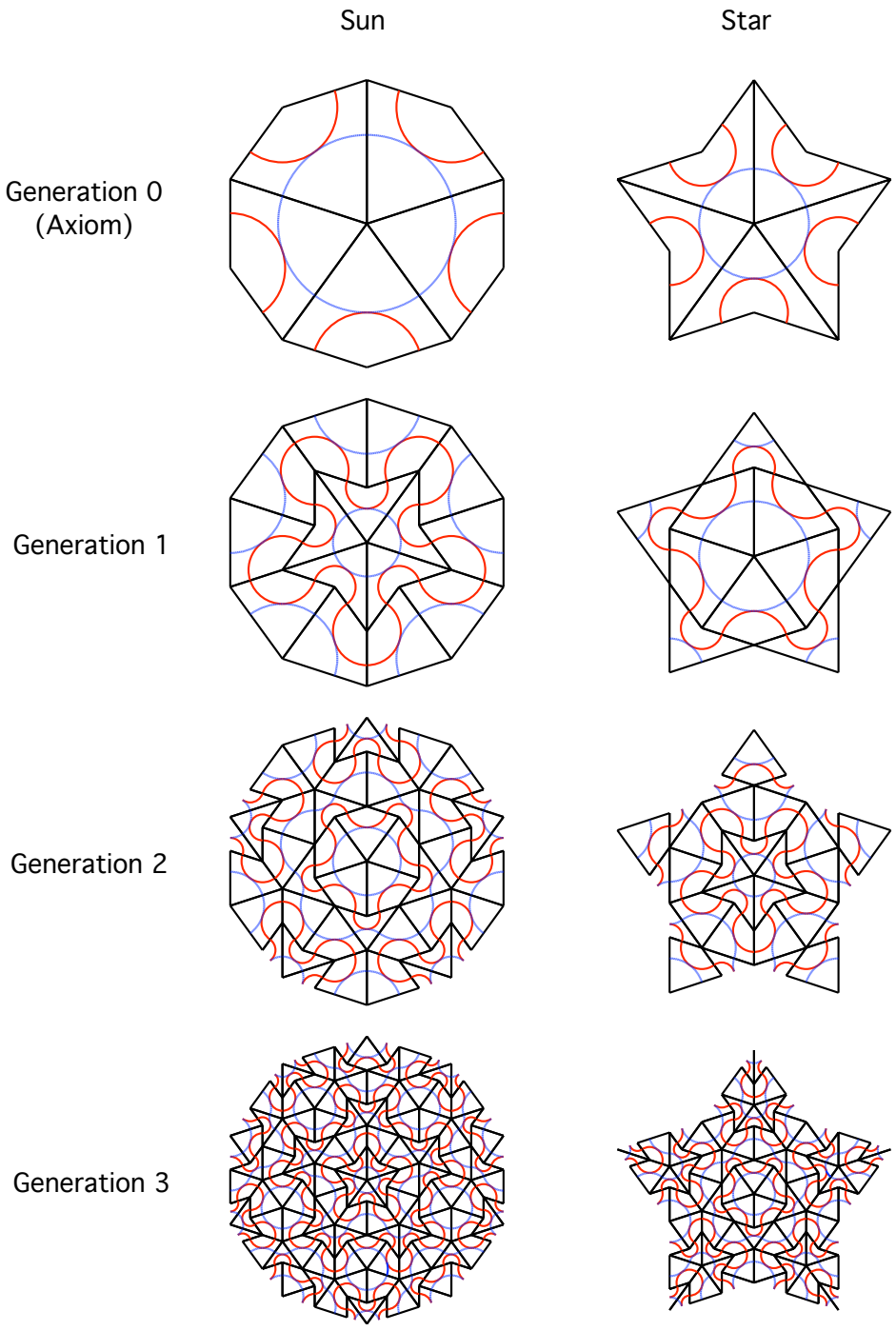


Figure 3.10: Four generations of Penrose tilings showing the process of deflation used to create them. The patterns start from either the 'Sun' or 'Star' axiom.

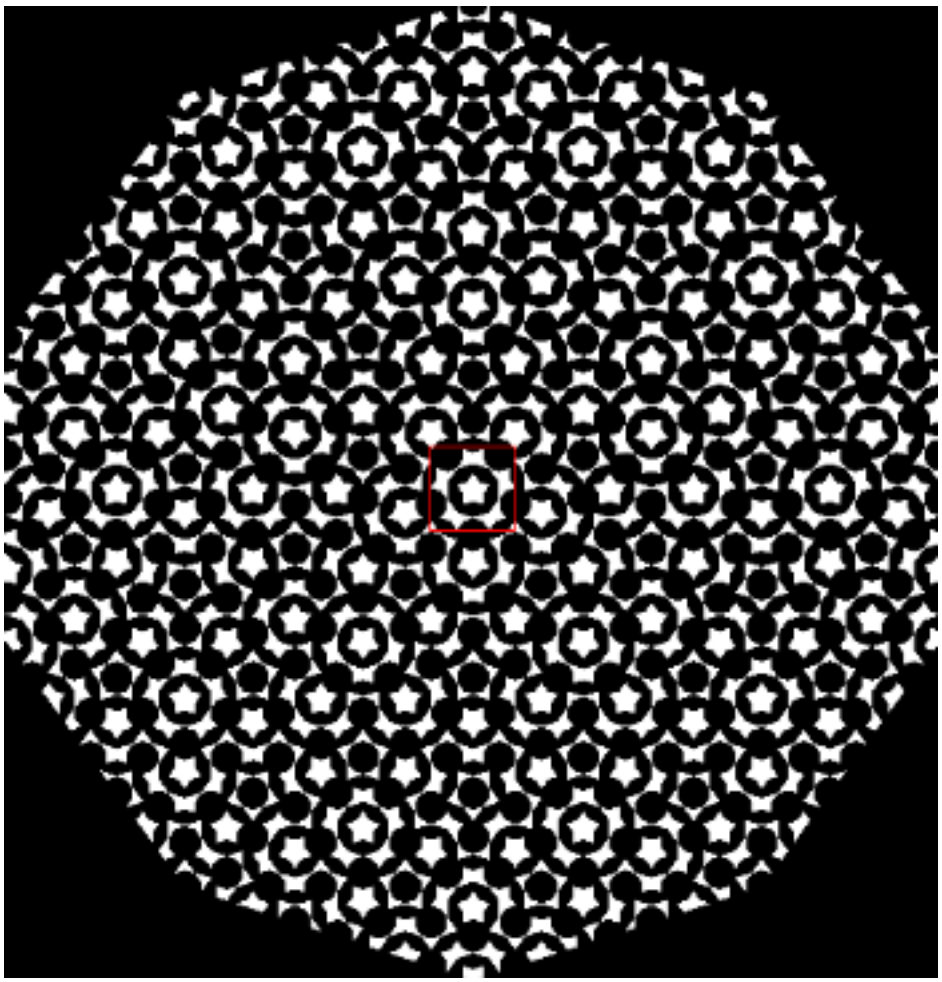


Figure 3.11: A seventh generation Penrose pattern based on the sun axiom, with the minimum interrogation area that produces a unique position when the correlation is performed highlighted.

3.2 Analysis

3.2.1 Simulation

To evaluate the various marker patterns introduced above several simulations were set-up allowing the process of correlation based alignment to be replicated. This has been achieved using a Java based image manipulation package developed by the National Institutes of Health called ImageJ [113], which has sophisticated support for user-defined macros and plug-ins facilitating the creation of image manipulation programs and has a large pre-existing library of image manipulation functions.

Using ImageJ has allowed simulation of the effects of noise, blurring and rotation on the marker position as measured by correlation to be investigated. It has also allowed simulations to be performed in which the accuracy of locating a numerically shifted marker could be measured. This has necessitated the development of several software routines to perform correlations on various types of images, to produce autocorrelation images and to give values for the PCE and other metrics related to the autocorrelation discussed below. The Java code is given in appendix A for three routines, which respectively compute the correlation of two images with periodic boundary conditions, calculate the peak-to-correlation energy ratio for a given correlation, and lastly compute the correlation and measures how the PCE varies as the values for blurring, sampling, rotation and noise functions are varied.

3.2.2 Metrics

Defining a metric to measure the 'goodness' of a pattern used in a correlation based alignment process is a non-trival task. Discussed here are the properties that make a pattern good for alignment and several methods of quantitatively measuring these parameters.

The design of the markers plays a crucial role in the accuracy provided by the correlation method and the best way to evaluate a given marker design is by examining the

autocorrelation of the marker pattern. This gives an indication of the resultant correlation in the ideal case when the image is a perfect match for the reference, i.e. the image is undistorted. The sharper the central peak of the autocorrelation the more accurately the centre of the pattern can be located and therefore the more precisely any misalignment can be measured using that pattern. It follows that to be able to locate a given pattern with a correlation method we would like the output of the correlation to have a significant value only whenever the reference pattern is visible in the field of view. Furthermore the output of the correlation should be sharply peaked so that small displacements of the pattern produce a significant change in the correlation output and hence the position of the pattern within the field of view can be located with a greater precision.

The autocorrelation of a pattern allows its suitability to be evaluated in the ideal case i.e. in the absence of sampling artefacts, pattern defects and other distortions. The autocorrelation is produced as described by equation 2.15 in section 2.2.4.2, so for a given pattern image, $p(\hat{\mathbf{r}})$, the autocorrelation, $A(\hat{\mathbf{r}})$, can be defined as follows:

$$\begin{aligned} A(\hat{\mathbf{r}}) &= p(\hat{\mathbf{r}}) \star p(\hat{\mathbf{r}}) \\ &= \mathcal{F}\left\{ P\left(\hat{\mathbf{k}}\right) \overline{P}\left(\hat{\mathbf{k}}\right) \right\} \\ &= \mathcal{F}\left\{ \left| P\left(\hat{\mathbf{k}}\right) \right|^2 \right\} \end{aligned} \tag{3.5}$$

This can then be evaluated to compare the performance of various types of pattern, however this requires that a figure of merit be chosen.

There have been a number of procedures used in the literature [37] to assign a measure of the quality of an autocorrelation; in particular a great deal of work has been done to quantify correlation filters for use in signal processing applications. Matched filtering, (see section 2.2.4.1), in communications systems also relies on the filter output being sharply peaked when the known signal is present in the detected waveform and being small valued

when only noise is present. This is similar to the autocorrelation properties desired for alignment and therefore it is anticipated that the performance measures developed in this field can be directly applied to image based alignment applications using correlation.

The first metric commonly used is the signal-to-noise ratio (SNR), which is a measure of the information content as related to the random noise carried by the output of the correlation. It is defined by equation 3.6, where $E\{\}$ denotes the ensemble average, $var\{\}$ denotes the variance and since $A(\hat{\mathbf{r}})$ is an autocorrelation, its maximum value will be at the point $(0,0)$.

$$SNR = \frac{\left|E\left\{A(0,0)\right\}\right|^2}{var\left\{A(0,0)\right\}} \tag{3.6}$$

The SNR cannot be calculated from a single autocorrelation output since the variance would be zero, but instead it is calculated by repeatedly performing the autocorrelation with a source of noise present. Schematically this is shown in figure 3.12. Note that the noise added to the input of the correlation is independent between the two inputs, in this case one input has zero noise and the other has noise $n(\hat{\mathbf{r}})$. Since correlation is linear this is the same as having two separate uncorrelated noise sources. From a group of correlations the SNR can be calculated using equation 3.6.

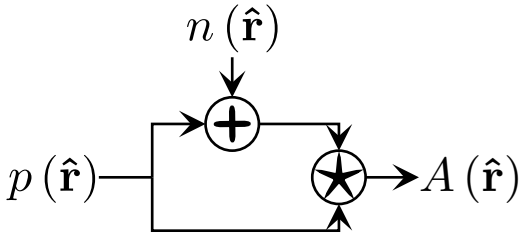


Figure 3.12: Flow diagram of the process used to measure the SNR of the autocorrelation, $A(\hat{\mathbf{r}})$, of a pattern, $p(\hat{\mathbf{r}})$, in the presence of a source of white Gaussian noise, $n(\hat{\mathbf{r}})$.

Closely related to the SNR is the peak-to-sidelobe ratio (PSR). Defined by equation 3.7, the difference from the SNR being that the variance of a point far from the peak

replaces the variance of the peak itself. The choice of τ_x and τ_y is arbitrary but by using a point around $\pm\frac{L}{2}$ when the correlation extends from $-L$ to L, gives a good separation from the peak without being adversely affected by any effects at the edge of the correlation (e.g. from the windowing function used, or due to the boundary conditions). It is important to note that in the case of only additive white Gaussian noise then the PSR and SNR give exactly the same measure.

$$PSR = \frac{\left|E\left\{A\left(0,0\right)\right\}\right|^2}{var\left\{A\left(\tau_x,\tau_y\right)\right\}} \tag{3.7}$$

From the discussion in section 2.2.4.1 it is evident that, in terms of SNR, correlation is the optimal process for identifying a known marker in the presence of additive white Gaussian noise. Therefore it is also the optimal process in terms of optimising PSR as well. As was discussed in section 2.2.4.1 the shape of the marker pattern has no effect on the optimisation of the SNR, Turin [97] simply states that, whatever the chosen pattern, matched filtering is the best method, in terms of signal-to-noise ratio, to locate the pattern amongst the noise.

The signal-to-noise ratio, and indeed the peak-to-sidelobe ratio, is an important performance measure when considering how robust a particular pattern will be to external interference, but as a measure of the theoretical performance for alignment it proves inadequate as a metric. The SNR (and PSR) considers only the average and variance of the correlation peak and not of any other points in the correlation and as such give no indication of the shape of the correlation peak. For alignment, the necessity of a sharply peaked autocorrelation has already been stressed and there are several so-called peak sharpness measures that aim to quantify the sharpness of the peak of an autocorrelation.

3.2.3 Peak Sharpness

Perhaps the simplest of the peak sharpness measures is the ratio of the correlation peak to the secondary peak. As the name suggests this is simply the ratio of the magnitude of the

main peak in the correlation to the magnitude of the second largest peak, as illustrated in figure 3.13. This is a relevant measure for alignment since it gives some indication of the how unique the main correlation peak is and gives some idea of the chance of the alignment algorithm mistakenly identifying a secondary peak (a false positive). It however, gives no indication of the spatial separation between the peaks or of the relative widths of the peaks and is therefore limited as a diagnostic tool.

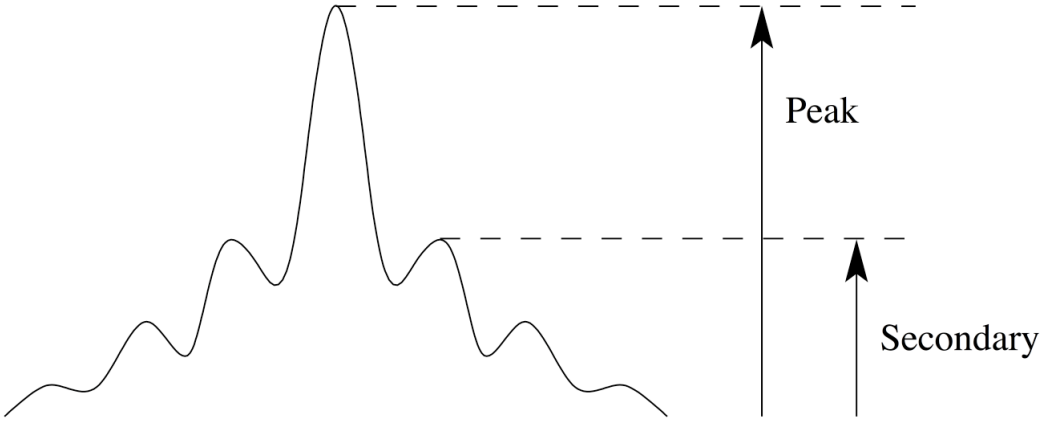


Figure 3.13: Illustration of the peak-to-secondary ratio for a 1-D correlation

Another measure that has been proposed to evaluate the peak sharpness is the peak-to-root mean square ratio or PRMSR. Defined as the ratio of the peak height to the RMS value of the pixels in the correlation whose values are below 50% of the peak value. This can be expressed mathematically as in equation 3.8 where Ω is the set of pixels with values less than half of the peak value and N_Ω is the number of pixels in that set. The biggest problem with this measure is the necessity of finding the right set of pixels to perform the averaging over, it does however give some idea as to how sharp the peak is since a sharper peak will have more pixels in the Ω set and will give larger values for the PRMSR. The weakness of the PRMSR measure is that it can only provide a meaningful measure of the sharpness of the central peak when it is the only significant peak in the correlation. When there are multiple peaks there could be a significant number of pixels

in the correlation with values greater than 50% of the peak, and thus ignored by the PRMSR.

$$PRMSR = \frac{\left|y\left(0,0\right)\right|^2}{y_{rms}^2} \tag{3.8}$$

$$where \qquad y_{rms} = \left[\frac{1}{N_\Omega}\sum_{i,j\in\Omega}\left|y\left(i,j\right)\right|^2\right]^{\frac{1}{2}}$$

The last of the peak sharpness measures is the peak-to-correlation energy ratio or PCE. This relates the energy contained in the central peak of the correlation to the total energy contained in the correlation plane. As such, it gives a direct measure of the sharpness of the central peak of an autocorrelation: a sharp peak will contain a larger proportion of the total energy within the central peak than a more widely spread out peak will. Mathematically it is given by equation 3.9, where $A\left(\hat{\mathbf{r}}\right)$ is again the autocorrelation.

$$PCE = \frac{\left|A\left(0,0\right)\right|^2}{\int_{-\infty}^{\infty}\left|A\left(\hat{\mathbf{r}}\right)\right|^2 \mathrm{d}\hat{\mathbf{r}}} \tag{3.9}$$

For correlation based alignment the most applicable of the peak sharpness measures is the peak-to-correlation energy ratio, since it measures the sharpness of the central peak, the peak used to locate a marker in correlation based alignment. The sharpness of the peak of the autocorrelation provides just one consideration when comparing marker patterns. There are practical limitations on the fabrication of the alignment markers, and the method used for interrogating the marker must also be considered to evaluate the suitability and performance of a given marker pattern.

3.2.4 Coarse Positioning

Coarse positioning is the process of determining the rough location within an extended pattern before the exact position can be determined with a much higher precision. It is

required because when a substrate is mounted on a holder and loaded into the machine, although the position is measured using an optical microscope with a stage equipped with linear encoders, the initial error in position can be several hundred micrometers.

The pattern used strongly affects the coarse positioning performance as the following example illustrates. Consider examining a small region from a large grating, since the region contains a large number of edges, the positions of the edges within the small region can be determined with a high degree of precision. However, there are many regions within the larger grating identical to the small regions examined and thus the position within the larger pattern remains unknown. The patterns examined above have two features that allow the general position within a larger pattern to be determine without interrogating the entire extent pattern. Firstly fractal patterns such as the Sierpinski carpet encode the long-range position by incorporating motifs defined on different length scales, thus the position within the complete pattern can be identified by interrogating a region containing a few differently sized motifs. In fact this is where ideal fractals excel since no matter what resolution is used for the interrogation the interrogated region will contain the same distribution of feature sizes. Therefore the same amount of information about the position within the entire pattern and the same amount of precise positional information can be obtained from the region.

There are several practical disadvantage to having both large and small structures within a marker pattern. Firstly the large structures produce dead-zones where there is a lack of high-resolution information. Taking the Sierpinski carpet of figure 3.6 as an example, the large central square in the pattern fills a large percentage of the pattern area. Within this are there are no edges and so if the interrogated areas lies within this region the marker's position cannot be determined. This means that while that region of the pattern allows the coarse position to be determined it does not contribute to the precise determination of the marker's position. Secondly it is difficult to optimise the fabrication process for both the large and small features within the pattern. Ideally the

smallest structures would be fabricated close to the resolution limit of the fabrication process, however the proximity effect in electron beam lithography makes it difficult to fabricate small features close to large ones. The proximity effect occurs due to the beam electrons scattering in the resist and backscattering from the substrate material. This results in the incident beam depositing significant dose in the resist over a large area ($\sim$20 μm radius but depends on the accelerating voltage and on the substrate material). This means that around a large pattern a large 'background' dose is deposited which limits the size of the smallest features that can be defined near to large features.

The second method that can be used to determine the coarse position within a pattern is by varying the local layout of motifs within a pattern. For example, since Penrose patterns are aperiodic, the motifs in the local neighbourhood uniquely determine the position within the complete pattern, and interrogating a given area of pattern uniquely determines its position within the full extent of the pattern. For an infinite Penrose tiling any region is repeated an infinite number of times. In the current case, where the pattern is finite in extent, there is a minimum area that must be interrogated, to uniquely determine the position. The advantage of this is that the motifs within a Penrose pattern are all similarly sized, and as such can be fabricated near to the resolution limit of the fabrication process, maximising the high-resolution positional information within a given area.

The problems of coarse positioning can be circumvented by using a two stage alignment strategy where separate coarse and fine alignment markers are used. This allows the individual markers to be optimised for each process. So in fact, although it is sometimes useful to have coarse positioning information within a marker pattern this is not essential and in certain cases may actually reduce the ultimate accuracy due to the reduction of high resolution information caused by the larger parts of the pattern, as was seen in the case of the Sierpinski carpet. A marker pattern that does incorporate coarse positioning information without a loss of high-resolution information, such as a Penrose

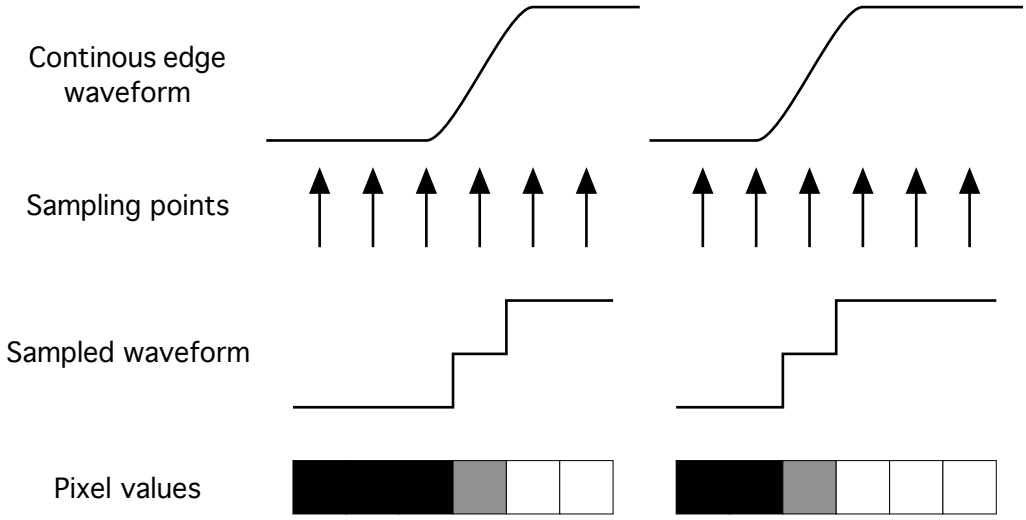
pattern, does offer the option of using a single interrogation to perform both coarse and fine positioning without reducing the accuracy of precision with which the marker can be located.

3.2.5 Grid Degeneracy & Sampling Effects

In practice images are represented as values on a discrete grid of points and, unlike the continuous plane, a grid of points has distinct anisotropy: directions parallel to the grid axes have special properties. Consider the sampling of a one-dimensional intensity profile across a pattern boundary as shown in figure 3.14(a). The edge can be seen clearly in the sampled intensity profile and by examining the relative changes in intensity between adjacent sample points the edge can be located to be at a specific pixel, i.e. by applying a threshold pixels with values less than the threshold are outside the marker while those pixels with values greater are inside. The shape of the sample intensity profile however, also contains information about the edge's position at a scale smaller than that of a single pixel. Imagine the boundary is moved by a whole pixel, the sampled intensity profile shifts correspondingly but there is no change in its shape, figure 3.14(b). If the boundary is shifted by a fraction of a pixel amount, figure 3.14(c), then the shape of the intensity profile changes showing that the shape of the profile encodes information about the sub-pixel position of the pattern boundary.

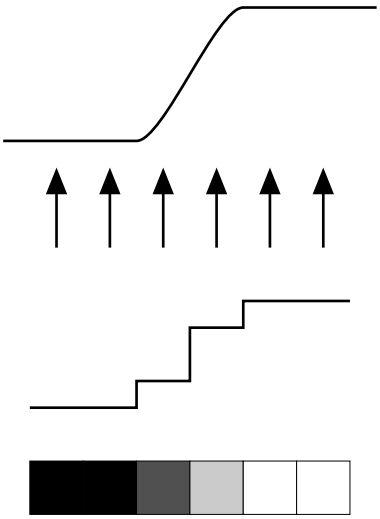
When the edge is over-sampled, such there are multiple pixels across each edge, the sub-pixel edge position can be determined with an arbitrary level of precision limited only by the shot noise of the detection system. This can be obtained since each of the pixels along the length of the boundary gives an independent measure of the edge position. However, when the edge is under-sampled such that the samples are either on the marker or off the marker, i.e. the edge slope is very much less than the sampling period, the precision of sub-pixel information that can be retrieved is limited by the pattern geometry and the sampling period.

Consider the example of figure 3.15, which shows a pattern with a vertical edge,



(a) Intensity profile of a continuous boundary after sampling

(b) Sampled intensity profile of the same continuous boundary after a single pixel shift



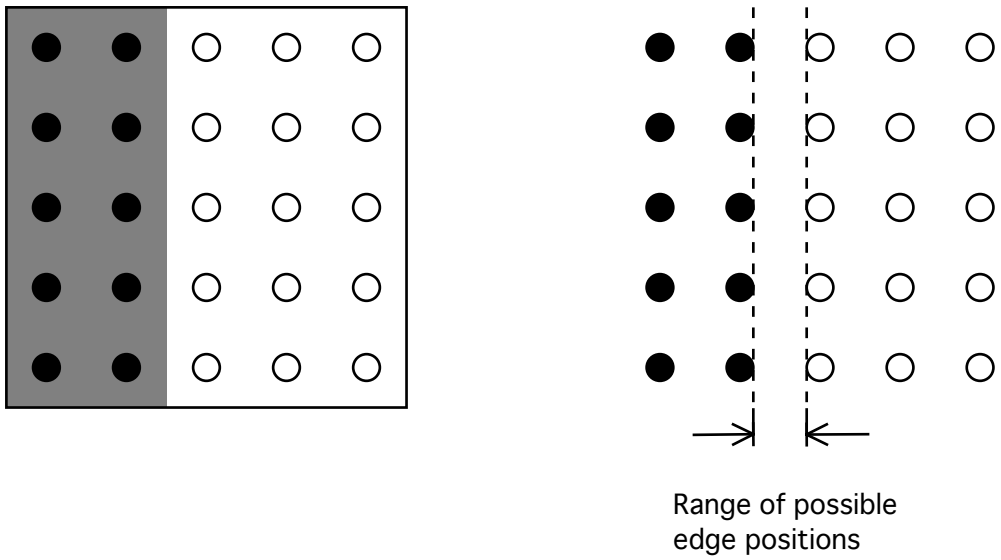
(c) Sampled intensity profile of the same continuous boundary after a sub-pixel shift

Figure 3.14: Effect of pattern shift on position and shape of its sampled intensity profile

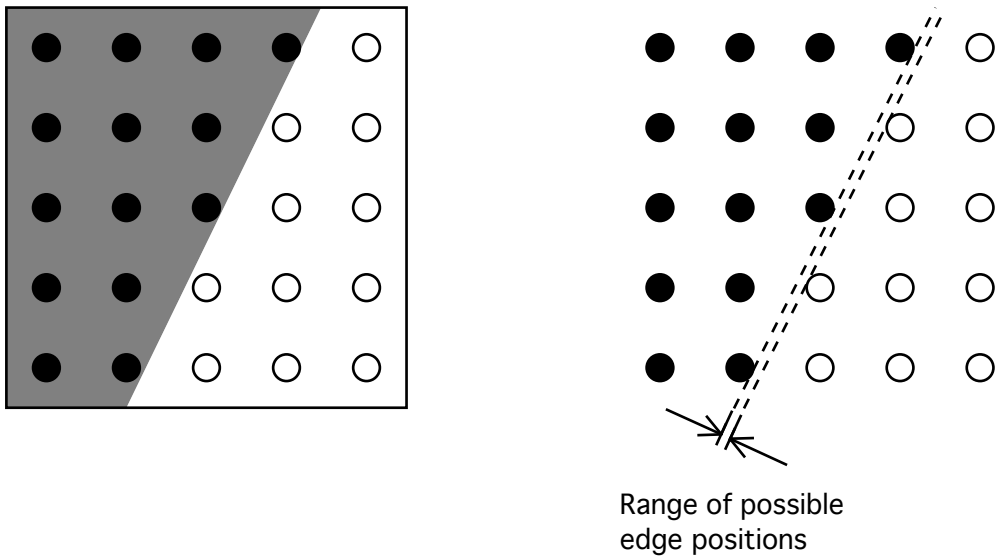
parallel to one of the sampling grid axes, and a pattern with an edge angled with respect to the sampling grid axes, sampled onto a grid of points with a period much larger than the edge slope. From the values at the sampled points of the vertical edge it is impossible to determine the edge's true position to the less than the distance between sampling points. The range of possible edge positions resulting in identical sampled values is demonstrated in the right-hand part of figure 3.15(a) by the dashed vertical lines. On the other hand, when the underlying pattern has edges that are not parallel to the sampling grid axes the uncertainty with which the edge's true position can be determined is greatly reduced. This situation is illustrated in figure 3.15(b), where again the dashed lines in the right-hand part indicate the, now much reduced, range of edge positions resulting in identical values at the sampling points.

There is a related effect dependant only on the pattern geometry and not on the sampling grid and relates to the sensitivity of a pattern to motion in a given direction. As has been already shown when an edge in the pattern lines up parallel to one of the sampling grid axes then the certainty with which the edge's position can be determined is reduced. There is a similar effect where small shifts of the pattern along a direction parallel to an edge result in no change in the sampled information for that edge. Considering only a single edge in the pattern this means that the positional information provided by a single edge in the pattern is dependant on the direction in which the pattern shifts. Therefore the distribution of edge directions in the pattern directly relates to the angular sensitivity to pattern shifts. So for instance all the rectilinear patterns have edges aligned to just two directions and therefore are more sensitive to pattern shifts in directions diagonal to the edges since in this case all the edges contribute equally to the measurement of position. For shifts parallel to either of the pattern edge directions only half of the edges contribute to the position measurement, and the sensitivity is reduced.

Examining the pseudo-random noise array pattern from figure 3.5 on page 61, it is clear to see that all the edges are parallel to the grid axes and that this pattern is



(a) Effect of undersampling a vertical edge onto a sampling grid and the range of possible edge positions that the sampled values could correspond to.



(b) Effect of undersampling an angled edge onto a sampling grid and the range of possible edge positions that the sampled values could correspond to.

Figure 3.15: Effect of pattern edge direction on the accuracy with which the edge position can be determined from the set of undersampled values.

severely limited by grid degeneracy. With a simple rotation of the entire pattern the pseudo-random noise array could be formed into a configuration where none of its edges are parallel to the grid axes, however the edges in the pattern are from just two directions and so the angular sensitivity is strongly influenced by this and when sampled the pattern is more sensitive to shifts in two angular directions.

The universal alignment target on the other hand is composed completely of curved boundaries which do not line up parallel to the sample grid axes for any significant length and also cover the complete range of directions in a much more uniform manner giving the universal alignment target good angular sensitivity over the complete 360^o. The Penrose pattern is somewhere in between these two extremes since it has edges in five distinct directions which gives it a fairly uniform angular sensitivity and a good degree of immunity to grid degeneracy since at most only a single direction, or $1/5$ of the edges, can be made parallel to a grid axis.

The marker pattern will always be undersampled since a relatively large area ($2\,\mu$m) must be imaged with a relatively small number of pixels ($\lesssim$100), as alignment must be achieved quickly. Since correlation is a linear process, the effect of undersampling is to introduce a background ripple in the correlation. The period of this disturbance in the correlation is equal the period of the sampling function. Since correlation is a linear process the undersampling artefacts can by removed by using a matched filter. The ripple amplitude has been found to be much smaller for Penrose patterns than for URA, as expected.

3.2.6 Fabrication Limitations

Since the focus is on creating a marker pattern to be used in ebeam lithography it is desirable to be able to create the markers also by ebeam lithography. This constrains the patterns that can be created in several ways. Perhaps most significant is the limit placed on the total area of the marker pattern. Ebeam lithography is relatively slow so due to time constraints it is desirable to fabricate markers with a relatively small

area. This is not just about saving time, but is also because the stage in an ebeam tool is constantly drifting to some extent and so writing the pattern quickly reduces any errors introduced by drift during the writing process. The limitation on area is also a benefit when considering the interrogation of the marker, since a smaller marker can be interrogated more quickly, also avoiding problems with drift.

As has been discussed, exposing patterns with large and small features in close proximity causes problems for ebeam lithography due to the proximity effect [114]. Variations in the size of exposed areas produce local variations in the background backscattered electron level which results in the size of patterned features being dependant on the local pattern density. A pattern which is composed of many similarly sized elements that are in a uniform density over the area of the pattern are therefore easier to fabricate accurately.

Markers were made using a lift-off process in which the minimum features that could be reliably reproduced are around 30 nm in size. This allowed a large process window in terms of dose and development to give the process some robustness against variations in either of these variables. To maximise the number of edges in the fabricated marker pattern it is beneficial to create as much of the pattern as possible at this minimum size, reinforcing the idea that the pattern should be completely formed of similarly-sized elements.

The design and fracturing processes in ebeam lithography necessitate the discretisation of the pattern data and therefore any curves can only be approximated by a staircase like edge, with step sizes defined by the pattern element size. This is also true for any sloped edges, i.e. edges that are neither horizontal nor vertical. This effect is reduced to a certain extent by the blurring effects of electrons forward scattering in the resist, the backscattered electrons contributing to the proximity effect and the limits on the resolution of the resist. However, any marker pattern must be tolerant of this effect. The universal alignment marker's potential for accurate alignment relies on its edges having distinct curvatures. It therefore must be written using small elements to preserve the

smoothness of the curved edges even though much of the pattern consists of large filled areas.

3.3 Pattern Choice and Justification

Each of the patterns discussed in this chapter has several features that make them in some way good for alignment. In this specific application, of an alignment marker for electron beam lithography, some of these features are more important. For the present application we wish to create a marker for use in electron beam lithography, preferably fabricated by ebeam, which will provide the highest possible accuracy from a correlation based alignment algorithm. Furthermore we propose to use the marker in a two-stage alignment process, as explained in the next chapter, chapter 4, where the first stage provides a good estimation of the marker position such that the second-stage of the alignment is used as a 'pull-in' to refine the alignment only. With these considerations Penrose tiling based marker patterns have been chosen as the optimal pattern for this work. This is due to their inherent aperiodicity that leads to a sharply peaked autocorrelation, giving a high level of accuracy when used in a correlation-based alignment process. Penrose patterns also exhibit $\sim$50% fill with similarly-sized elements which maximises the high spatial frequency components while also allowing the pattern fabrication process to be optimised easily. Lastly, the incoherence with a rectangular sampling grid makes it less susceptible to problems of grid degeneracy and of sampling.

Within the family of Penrose patterns there are an infinite number of possible sections that could be used as marker patterns. Based on the discussion in section 3.2.6 above, the chosen pattern has been limited in extent to 10 μm, roughly the same as for conventional simple geometric shaped cell markers. To give features within this pattern that are close to the fabrication limit a 7^{th} generation pattern has been used. To restrict the pattern to as compact an area as possible to avoid using more space on the substrate a Sun based pattern has been chosen. A SEM image of the resulting marker pattern as fabricated is

shown in figure 3.16.

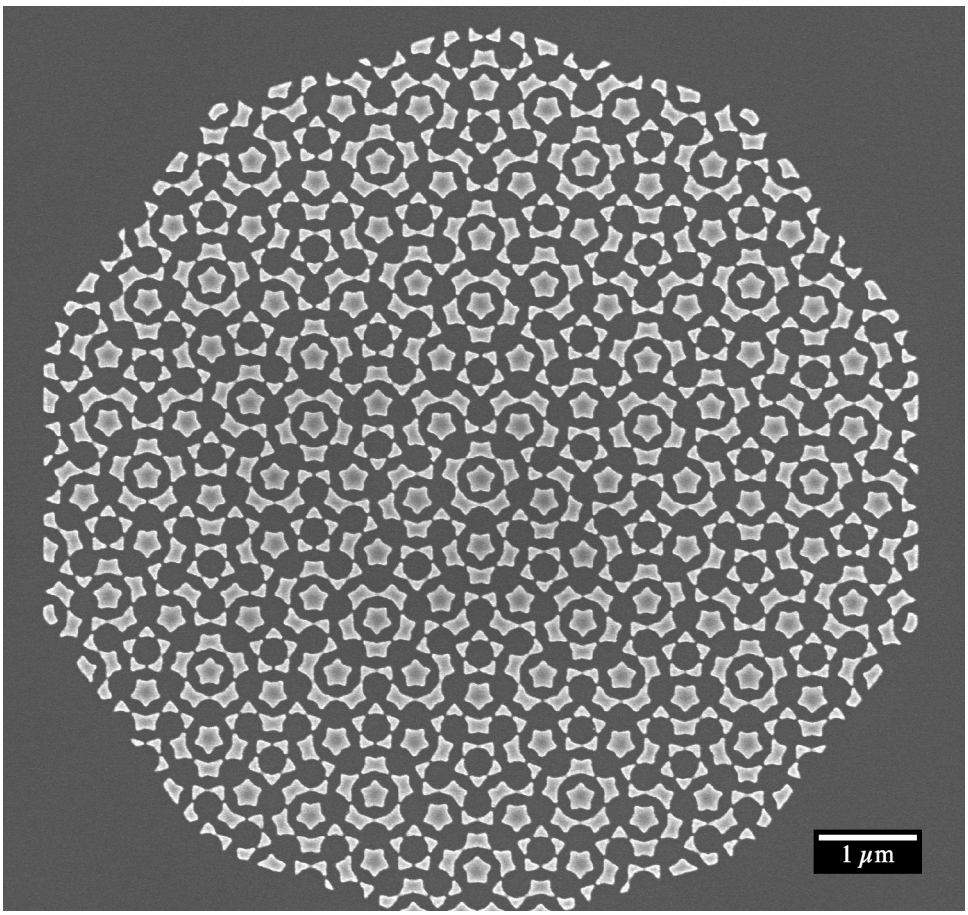


Figure 3.16: SEM image of a fabricated seventh generation Penrose pattern based on the Sun axiom with darts selected to be marker elements. This may be compared with the design shown in figure 3.11.

In the next chapter the specifics of implementing a correlation-based alignment process using Penrose pattern markers will be discussed.

4 Implementation

The process of implementing an alignment scheme based on correlation and using Penrose pattern alignment markers can be divided into three distinct tasks; the development of software to control the electron beam lithography tool and to carry out the correlation algorithm, the design and physical fabrication of marker patterns based on Penrose patterns and thirdly a validation process to verify the operation of the system and demonstrate its performance. As part of the validation a method of reliably measuring alignment on the nanometer scale will be introduced and its use in verifying the improvements the correlation-based alignment with Penrose pattern markers provides will be discussed. Following that, the performance of the correlation based system with respect to the conventional alignment method will be examined. Each of these topics will be dealt with in turn in the following sections.

4.1 Software

The VB6 ebeam machine, as described in section 2.1.1, is largely controlled from a Linux workstation. Vistec supply a set of software tools which act as a control system for the tool. This system, called EMMA [115], allows the machine to be controlled simply by issuing commands at a shell prompt much like running any program under linux. This system allows all the functions of the tool to be controlled, from loading substrates from the loadlock onto the stage, performing diagnostic functions, through to exposing pattern data and performing alignments.

It is useful to understand the flow of operations needed to expose a substrate with a given pattern. This is split into two parts; data preparation, which was described in section 2.1.1.8, and sample preparation which will now be discussed.

Assuming that the substrate has already been cleaned and that the resist has been spun and baked, the first task is to mount the substrate on a holder. A picture of the 4" wafer holder is shown in figure 4.1, in which, the main components have been annotated.

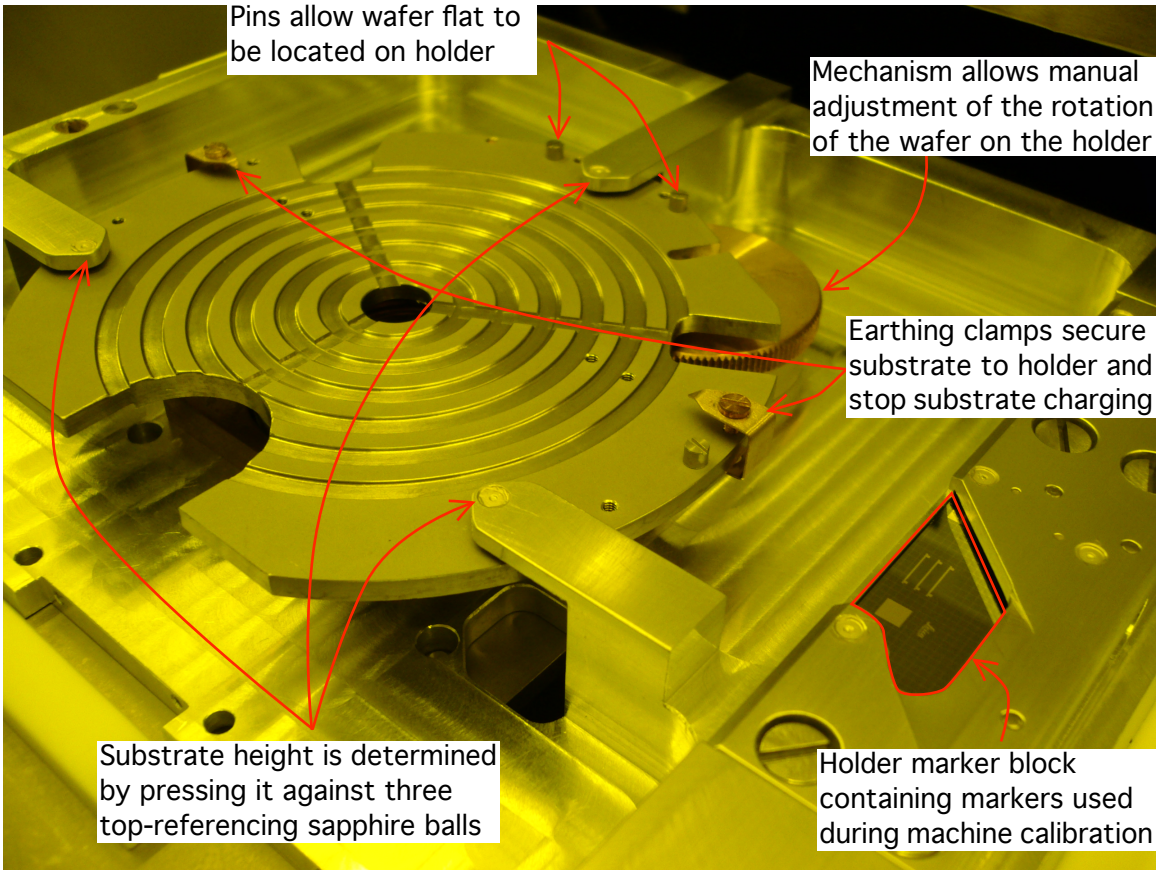


Figure 4.1: Annotated picture of the 4" wafer holder, showing the main components of the holder.

To load the wafer an operator physically slides the wafer into the holder until the flat rests against the holder pins which helps to orientate the substrate. The spring-loaded holder plate is lowered to allow the wafer to be slid into place and is then released which

pushes the wafer up against three sapphire balls which ensure that the top surface of the wafer is at the correct height, a technique known as "top-referencing". Using an optical microscope, frequently referred to as the "alignment microscope", the rotation of the wafer is checked with respect to any pattern already on the wafer. The alignment microscope's stage is fitted with a linear encoder to measure its position and this is set-up to read the same positions as the VB6 stage would. Using features on the wafer that are optically visible the rotation is adjusted by the operator to be better than $3\,\mu$m/ mm (3 mrad), as this is maximum rotation that can be corrected electronically by the deflection system. While the holder is under the alignment microscope a note of the position of the cross on the wafer is taken, this allows the same cross to be found once the holder is loaded onto the stage in the VB6. Finding the cross is the first step in the alignment process as it allows one physically measured point on the sample to be linked with the stage position when the cross is viewed in SEM mode on the electron beam lithography tool.

Mounting parts of wafers is slightly more complex, in that they are clamped onto a plate with an array of tapped holes known as a "piece-part" holder. Figure 4.2 shows the piece-part holder mounted under the alignment microscope, with two small samples mounted on the holder.

These pieces are not top-referenced, so to ensure that their surface is at the correct level the plate in the holder is mounted on three screws so that the height and tilt of the plate can be adjusted. The alignment microscope also contains a heightmeter set-up to measure the substrate height in the same way as the VB6, this allows the height and tilt of the substrate to be checked and adjusted. The rotation is adjusted by manually nudging the substrate and again a note is made of the position of the substrate so that it can be found once the holder is mounted on the VB6 stage.

Once the substrate has been mounted on the holder it is placed inside the loadlock and then the loadlock is evacuated. A robot arm inside the machine then moves the

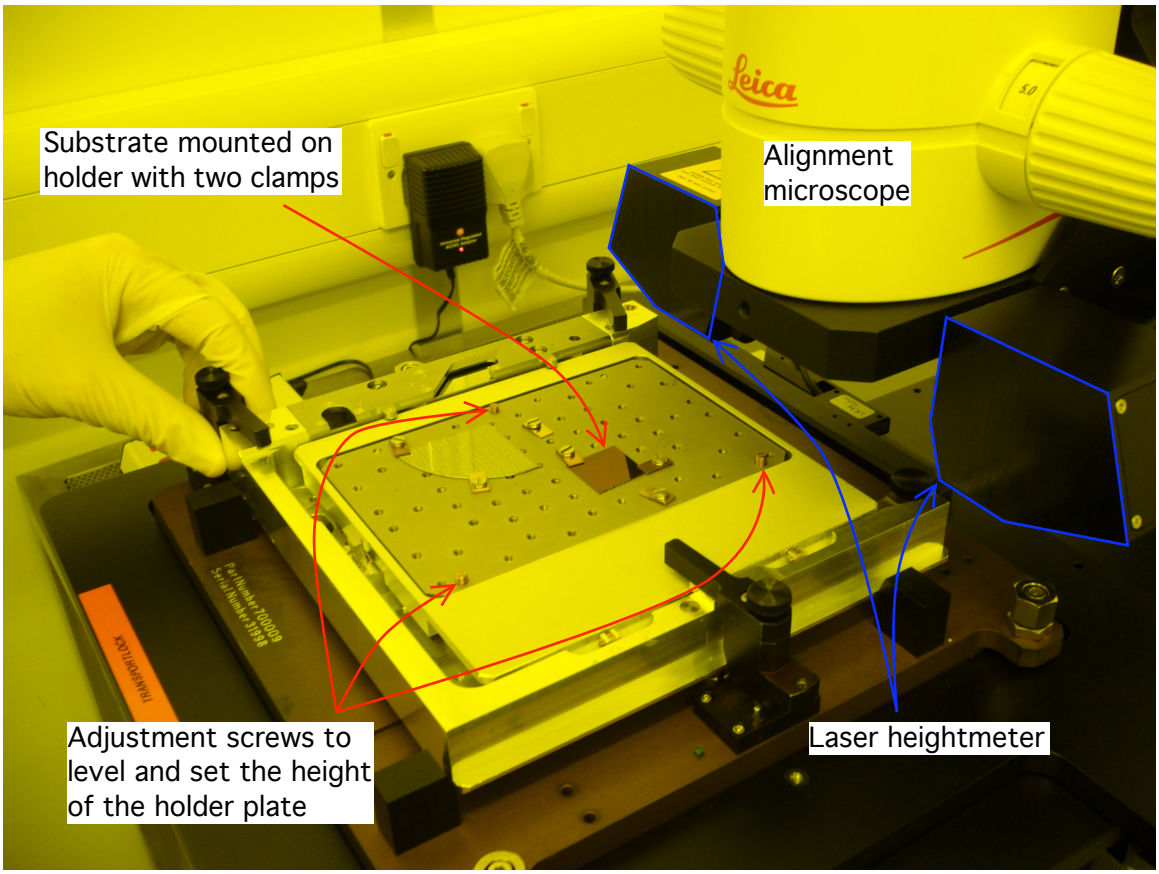


Figure 4.2: Annotated picture of the piece-part wafer holder, mounted on the alignment microscope.

holder from the airlock and mounts it on the stage of the machine. Using the noted position of the cross or the substrate's corner the same point is located using the SEM mode of the VB6 and the stage co-ordinates are stored as the starting position for that job.

Traditionally when exposing a pattern with an ebeam machine, a single job file is created which contains all the jobs parameters. However, at the University of Glasgow there is a locally written software program, called "Runjob" [85], which automates the process of writing users' jobs so they require no human intervention at run-time. This software effectively works out the commands that need to be issued to the EMMA control software, the order in which they need to be processed and intelligently deals with any responses from the machine to cope with errors. The main advantage to having an automated software system to control the exposure of jobs is that it allows a much higher throughput to be achieved compared to a machine that needs to be user controlled during the exposure of a job. This is because several substrates and wafers can be loaded into the machine at one time. Following a set-up period these substrates can be sequentially exposed automatically with no further user intervention, a process which is typically undertaken overnight at Glasgow.

Since the alignment process is integrated into this program, it is necessary to spend some time explaining the role of Runjob. Figures 4.3-4.5 show simplified flow diagrams of the main logic in the Runjob program. (No attempt has been made, in figures 4.3-4.5, to detail the error handling routines in Runjob.)

To implement a correlation-based alignment scheme on the VB6 ebeam machine a new series of software routines are needed. These routines, which perform the correlation-based alignment, are added into Runjob in place of the standard marker search and interface with the machine hardware in the same way through EMMA. The correlation-based alignment routines replace the "Search for marker using parameters given by settings" functions within the "Find markers" sub-routine in figure 4.5 (the function box replaced

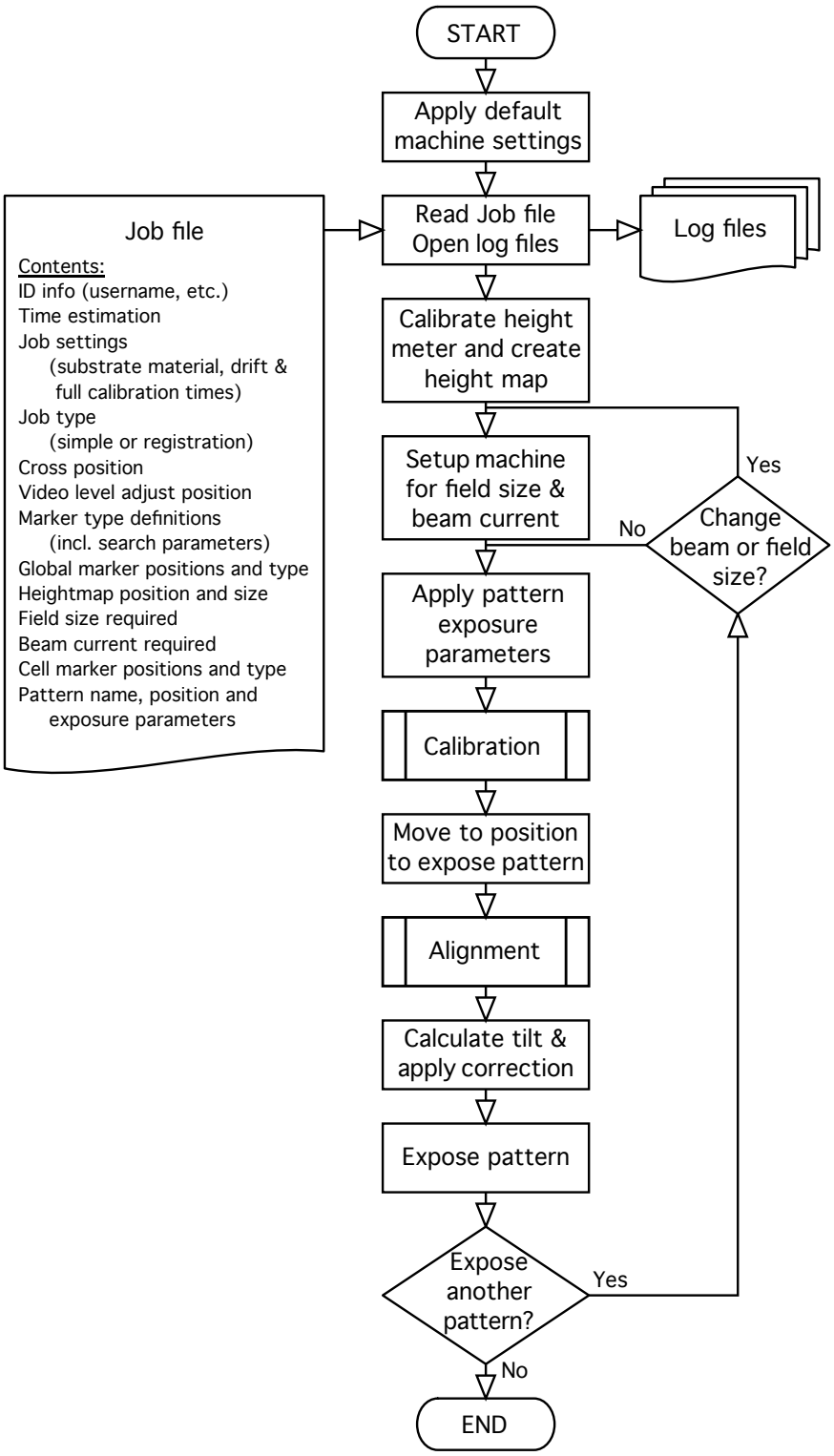


Figure 4.3: Flow diagram showing the main logical flow of the "Runjob" software. See figures 4.4 & 4.5 for details of the "Calibration" and "Alignment" routines.

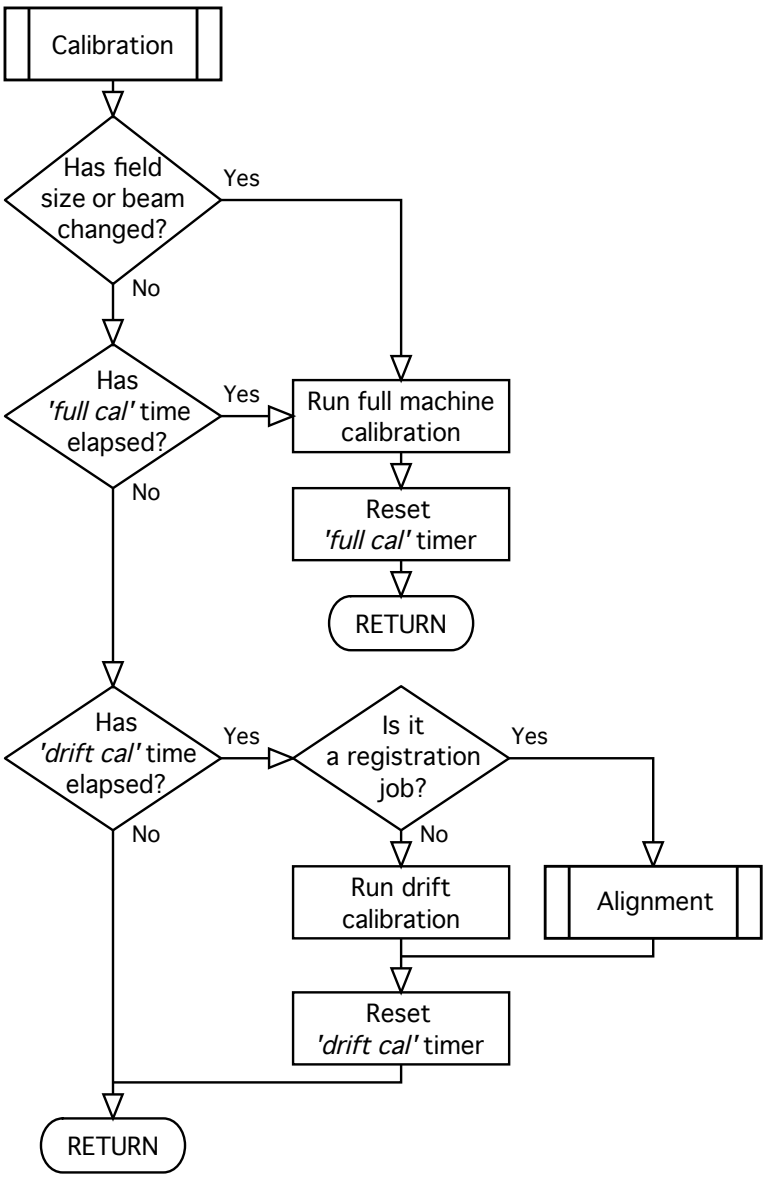


Figure 4.4: Flow diagram showing the calibration procedures used in the "Runjob" software.

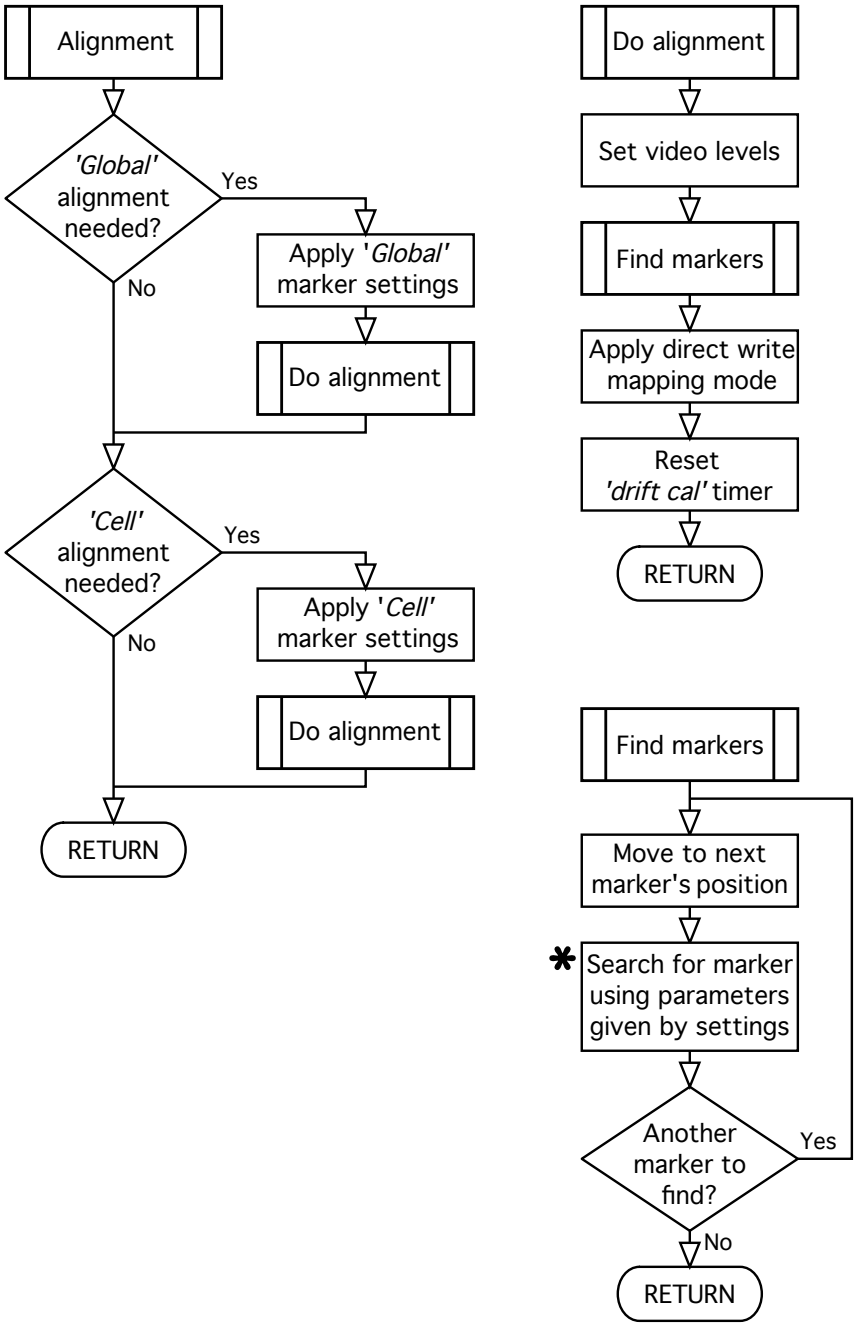


Figure 4.5: Flow diagram showing the routines used to perform alignment in the "Runjob" software.

by the correlation based alignment is indicated by the asterisk).

This function requires that the stage has already been moved and the deflection set such that the beam is positioned over the expected marker position, to an accuracy of $\pm 500\,\mathrm{nm}$. It also requires that the marker parameters and markers search settings from the job file are passed to the function. The function performs the marker search, as described in more detail in the following paragraphs, and the only thing it is required to return to the rest of the Runjob program is a position symbol containing the position at which the marker has been located or an error to say that the marker could not be found. In principle the details of the actual marker search are irrelevant to the rest of Runjob and it is this that allows any number of different methods to be used to locate markers in a straight forward fashion.

The "Search for marker" function calculates the actual marker position returning the four actual positions as it exits from the "Find markers" function. The software sends these positions to EMMA, instructing it to calculate the required mapping transformation coefficients and to produce a transformation to convert the ideal positions in the design file to these actual positions. Once the mapping transformation has been created, the EMMA software is then directed to update the internal systems on the electron beam machine and apply the transformation to all future stage movements and beam deflections.

Therefore to fit in with the requirements of Runjob the correlation-based alignment software has three main tasks to perform: firstly, it has to collect an image of the region around the expected marker position. Secondly this image needs to be correlated with a pre-defined reference image. Third and lastly, the marker position must be extracted from the correlation and returned to Runjob.

To collect an image of the marker area, use is made of the EMMA 'image collection' command, which allows a raster scanned image to be collected and stored in a file. The command has a range of parameters which allow the size, resolution, filter settings and sampling conditions to be controlled. Once the command has been executed the image

data points are saved in the form of a binary file where each byte of the file encodes the video level at each pixel. A text file accompanies the binary file and this contains a description of the contents of the image file. The text file records the position of the image, its size and the spacing of the sampling points used (i.e. the pixel size). As an aside the pixel values are in effect the value of the backscattered electron signal at a point on the substrate and are not extended objects with size equal to the pixel spacing. That said however, the electron beam has a finite size and so each pixel's intensity is formed from a region on the sample approximately equal to the spot size. The correlation software issues the EMMA command to collect an image and then, using the description file, reads the contents of the binary image file into memory so that the image can be further processed by the software.

When collecting an image it is important to consider the effects of the imaging conditions on the correlation process to avoid or reduce any effects that are detrimental to the correlation process. The relevant options which can be changed are:

1. Size of image - The area interrogated. The width, w, in the x-direction and height, h, in the y-direction, measured in μm.

2. Number of pixels - The number of pixels in the image in the x & y-direction, N_x & N_y respectively.

3. The filter time constant - Time constant, t_f, of the RC filter used for noise reduction, measured in μs.

4. Number of point samples - Number of times each pixel is measured and averaged to get a pixel value, S_p.

5. Number of full frame samples - Number of times the whole image is collected and averaged, S_f.

The requirements on the image size and number of pixels are linked and are somewhat constrained for two reason; there is a certain area of the defined marker pattern that must be interrogated to allow the position within the pattern to be determined uniquely, as discussed in section 3.1.7. Going beyond this size more information is gained and so the position of the marker can be determined with less susceptibility to noise. However to get this extra information the pixel size must remain the same, i.e. the number of pixels must increase. Increasing the number of pixels increases the processing time required to collect and correlate the image and it is of utmost importance to minimise this time, considering that many hundreds of marker searches could be required during the exposure of large substrates.

The other three parameters, filter time constant, point samples and frame samples, essentially provide three distinct methods to allow noise to be reduced during the image capture process. The image capture process is subject to shot noise from the beam as discussed in section 2.2.1.

The first, filtering, applies an analogue RC low-pass filter to the signal coming from the backscattered electron detectors, the time constant of this circuit being variable, to allow a greater or lesser effect. The low-pass RC filter introduces a delay in the measurement of the BSE intensity at each pixel. Since there is always drift present in the ebeam machine any delay between measuring the BSE intensity at consecutive pixels introduces some unwanted positional shift. However, the current produced by the semiconductor detector does not instantaneously change value, nor does the output of the electronic read-out circuitry change instantaneously. Thus, there is a "settling time" required for the detector and read-out circuitry to produce a meaningful output. A trade-off has to be made between speed of interrogation and accuracy of measurement.

Point sampling determines how many times the video level is measured at each point, the final value being the average of all the samples. Each sampling occurs after the RC filtering of the video signal has been performed and the output signal has stopped varying

due to settling issues. The point samples are therefore taken in succession separated by a period of time equal to the RC filter time constant, t_f. They are then averaged to reduce the effect of any time varying noise sources, such as the shot noise in the beam. Frame sampling works in a similar way although instead of repeatedly measuring each pixel and averaging, multiple complete images are collected and averaged to give the final output. Due to the fact that the stage is always drifting in an ebeam machine, frame averaging can introduce artefacts into the image due to the fact that the frames averaged together are not perfectly aligned because of this stage drift. This results in a blurring of the image and the edges of the features within the image become less well defined. The effect on correlation, when the image is blurred, is to reduce the sharpness of the correlation peak, since there will be a range of positions that match the reference image equally well. This in itself is not a problem since the centre of the peak will remain unshifted and the markers position will be calculated correctly giving the average marker position over the sampling period.

In the case of point sampling, the stage drift introduces some positional shift between pixels in the image and this introduces a stretching distortion into the image. If the stretching of the image can be assumed to be linear across the image then the effect is to widen the peak of the correlation. This is because the displacement between reference and image that provides the best match varies as the local area within the image varies. So the displacement that gives the best match for the top-left corners is not the same as for the bottom-right corners and this results in the correlation peak being spread out. When the stretching is non-uniform however, the peak in the correlation will also be affected in a non-uniform manner such that the peak will no longer be symmetrical. This causes errors in the calculated marker position when sub-pixel information is extracted from the correlation. Therefore in most cases it was felt to be safest to use frame averaging as a noise reduction technique since the distortions introduced into an image by frame averaging do not affect the position of the correlation peak and hence do not affect the

calculated marker position. Point sampling and large filter time constants were avoided, since they can introduce non-uniform positional shift across the image and affect the position measurement from the correlation peak.

Once the image has been collected and loaded, it then needs to be correlated against the reference image. Several options exist for generating the reference image. Perhaps most obvious is the use of a reference image derived directly from the pattern file used to create the markers. There are a few complications with this method however. Firstly, it is crucial to ensure that the digital versions of images have the correct dimensions and pixel sizes with respect to the physically collected images. This is important since any differences here could create scale changes for the correlation to deal with. Secondly, the markers are never an exact replica of the pattern data; the exposure and pattern transfer processes all distort the pattern in some way, reducing the high spatial frequency components, such as sharp corners and edges, and often results in some size bias between the fabricated structures and the designed pattern. Thirdly, the image collected by the VB6 will never compare exactly with the digital representation because it is always sampled by a spot with significant size, thus the image is blurred by the beam profile.

The second and simpler option is to use another collected image. The advantage of this method is that it is quick and that the reference and the collected marker images are both formed in the same way, with the same contrast mechanism, and same distortion. Therefore using an image of a fabricated marker as a reference compensates for these effects during the correlation.

The disadvantage is that the correlation provides the point of best match between the two input images, a relative measure, and not the absolute position of a point in the marker pattern. This means that multiple alignments to the same marker would be referenced to the same point but that that point is not necessarily the centre of the marker pattern. This is not a significant problem when the typical process flow for creating very small structures by correlation-based alignment is considered, see chapter 6.

For practical reasons an image of a marker was used as a reference image and the use of a marker generated from the pattern data was not investigated.

With the image collected and the reference image loaded the second task the software must perform is the correlation of the image data against the reference data. To do this equation 2.12 must be rewritten for the special case of discrete data. In continuous 2D space the correlation of two functions, $f(x,y)$ and $g(x,y)$, is given by

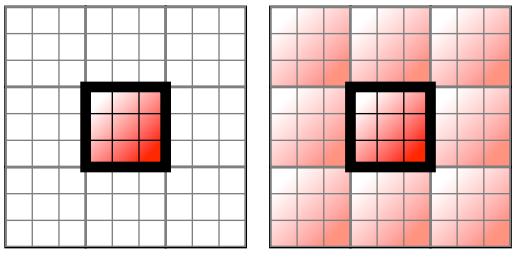
$$\begin{aligned} c(x,y) &= f(x,y) \star g(x,y) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(k,l)\, g(x+k, y+l)\, \mathrm{dk}\, \mathrm{dl} \end{aligned} \tag{4.1}$$

In 2D discrete space this can be expressed as

$$\begin{aligned} c(x,y) &= f(x,y) \star g(x,y) \\ &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f(k,l)\, g(x+k, y+l) \end{aligned} \tag{4.2}$$

Although the limits of the integral and sum are $\pm\infty$ to allow for the mathematical possibility of functions of infinite extent, in reality the images used are finite in size and care must be taken as to how the edges of the images are treated. The boundary conditions used define the extent of the correlation providing meaningful positional information. The two most commonly used boundary conditions are zero-pad and periodic extension. Zero-pad fills all space outside the images with zero value, periodic boundary conditions tile the image to form a periodic pattern in both dimensions, see figure 4.6.

Using the graphical interpretation of correlation, represented in figure 4.7, it can be seen that the value of each pixel in the correlation corresponds to multiplying together the overlapping pixels of the two images to be correlated, where the offset between the



(a) Zero-padding (b) Periodic extension

Figure 4.6: A 3x3 pixel image extended to 9x9 pixels using zero-pad or periodic boundary conditions.

two images is determined by the position of the pixel in the correlation. Figure 4.7 shows the case when the images are extended by zero-padding. There are 5x5 points with non-zero value in this case, for all other points in the correlation the relative offset of the two images means that they do not overlap and so the correlation has zero value.

In the case of periodic boundary conditions there are no zero value regions which can be ignored in the correlation, however, because of the periodicity of the situation it is still relatively straightforward to calculate the correlation. Figure 4.8 shows pixels (1,1) and (4,4) of the correlation in more detail. This time periodic boundary conditions have been applied to the images but for clarity this is only detailed for the yellow image. The pixels of the images have also been sequentially numbered from 1 to 9 to allow them to be more easily identified.

From figure 4.8 it can be seen that the value for the correlation pixel at these two locations must be the same, and that therefore for the case of periodic boundary conditions the correlation is completely defined by just 3x3 pixels. Therefore careful choice of boundary conditions is not only essential for correctly interpreting the correlation results but can also reduce the number of calculations needed. Several algorithms also exist to improve the efficiency of correlation calculations, most notably those based on Fourier transforms. In a similar way as for convolution, calculating the correlation can be done as a multiplication in the Fourier domain, as noted in sections 2.2.4.2 and shown in

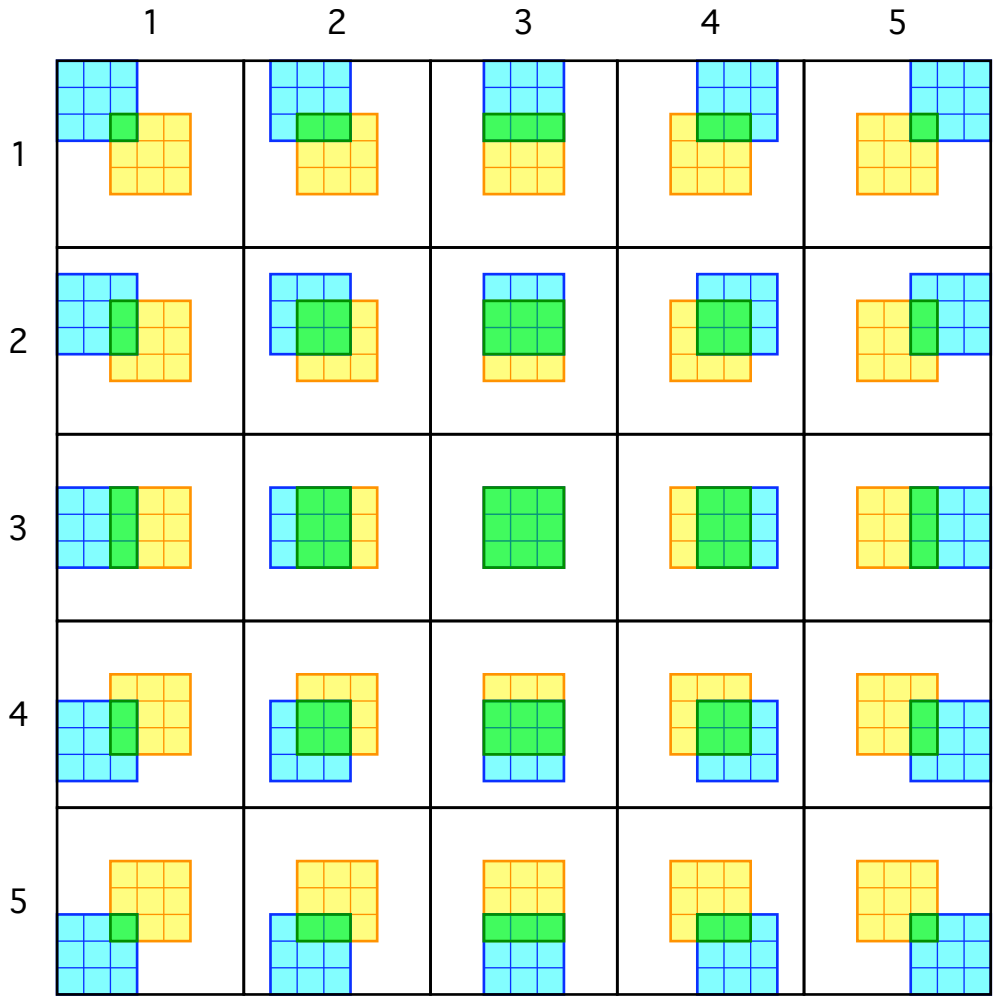


Figure 4.7: Graphical representation of correlating two 3x3 pixel images, shown as the yellow and light blue grids, to give a 5x5 pixel correlation, shown as the large grid. The value of each pixel in the correlation, is calculated by multiplying together the overlapping pixels in the two images to be correlated, the overlapping pixels are shown as green in the picture. The relative displacement of the two images is determined by which pixel in the correlation is being calculated. Zero-pad boundary conditions have been used.

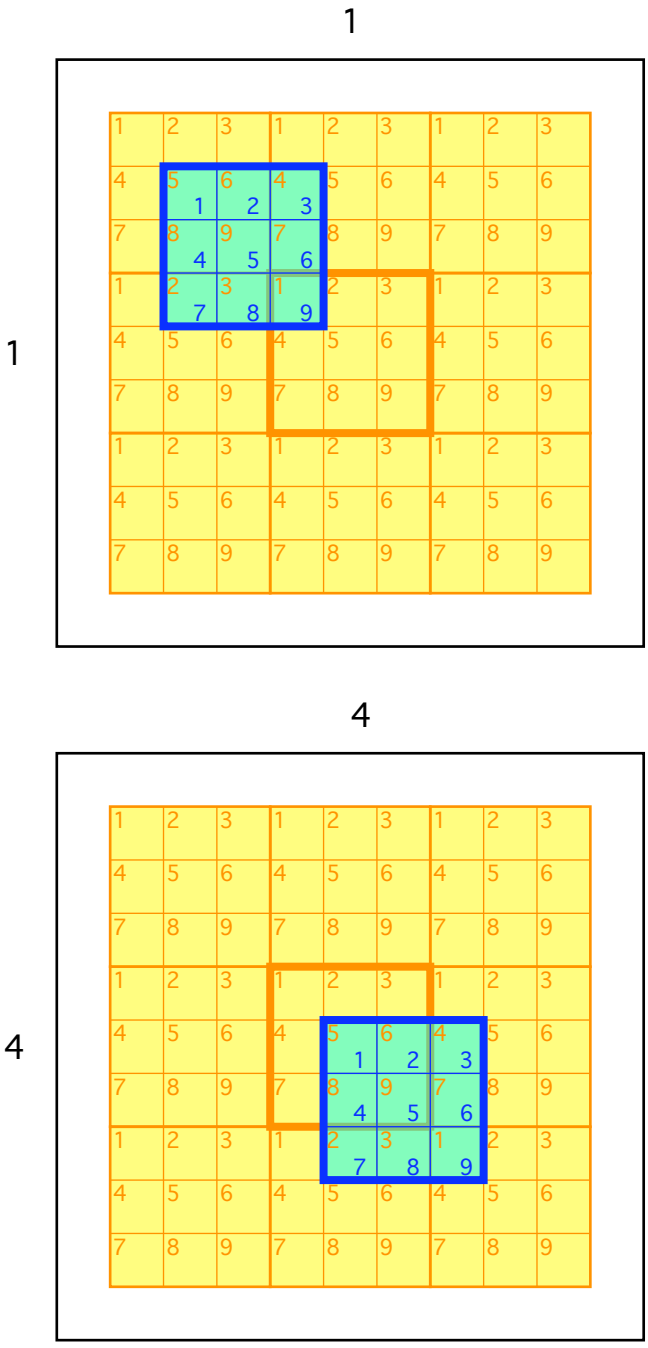


Figure 4.8: Detailed graphical representation of two pixels of the correlation of figure 4.7 when periodic boundary conditions are used. For clarity, the periodic extension of only one of the images is shown and the original images are identified by the thicker outlines. The image pixels are numbered for ease of reference. The image shows that the two pixels of the correlation detailed have the same value and shows that with periodic boundary conditions there are only 3x3 distinct pixels in the correlation.

equation 4.3.

$$f(\hat{\mathbf{r}}) \star g(\hat{\mathbf{r}}) = \mathcal{F}\left\{F\left(\hat{\mathbf{k}}\right) G^*\left(\hat{\mathbf{k}}\right)\right\} \tag{4.3}$$

This involves taking the Fourier transform, typically with a Fast Fourier Transform (FFT) algorithm, of each of the two images, multiplying the resulting images and doing an inverse Fourier transform to obtain the result. The speed improvement is only significant when images with large numbers of pixels are to be correlated. If N_p is the number of pixels in the image to be correlated, then $\mathrm{O}\left(N_p^2\right)$ calculations must be performed to obtain the correlation. Using FFTs this is reduced to $\mathrm{O}\left(6N_p \lg N_p + N_p\right)$, where lg is the base-2 logarithm.

Once the correlation has been calculated the software must have a method of locating the peak of the correlation, since the position of this peak corresponds to the offset of the marker. Section 2.2.4.2 reproduced the proof that the autocorrelation has its maximum at $(0, 0)$ and also stated that when cross-correlating an image with a shifted version of itself, then the peak of the correlation is shifted by the same amount. This was for the continuous domain however. Sampling of the correlation does not affect the peak position so the same results hold for the discrete domain. Thus, to improve the accuracy of the measurement of the offset, interpolation can be used on the discrete domain correlation, as a measurement of the underlying continuous function, so that the peak can be located with sub-pixel accuracy.

This was achieved by performing a least squares fit on a section of the correlation directly around the peak to a two-dimensional Gaussian function using the Levenberg-Marquadt non-linear least squares fitting algorithm [116–118]. A two-dimensional Gaussian function has been chosen since central limit theorem would suggest that the peak of the autocorrelation will tend towards a Gaussian function in shape. The Levenberg-Marquadt algorithm is an iterative process that interpolates between the Gauss-Newton algorithm and the method of steepest descent to minimise a non-linear function. It

has the advantage of improved robustness over the Gauss-Newton algorithm, especially with poor starting conditions but at the expense of being slower. Like the steepest descent method the Levenberg-Marquadt algorithm will always converge however it is faster then the steepest descent method. Thus the Levenberg-Marquadt algorithm can robustly find a minimum of a non-linear function with reasonable speed [93]. Fitting the two-dimensional Gaussian function to the 5x5 pixel region around the peak of the correlation allows the peak position to be established with sub-pixel precision.

4.2 Fabrication of Markers

In micro and nano fabrication and in particular in electron beam lithography it is relatively uncommon to have a level of lithography which just defines the markers. They are frequently written at the same time as the first level functional structures, such that the subsequent levels of lithography can be aligned to that first level. This procedure compromises the formation of the alignment markers to fit in with the requirements of the functional structures.

Fabricating markers in a separate step to the rest of the device is advantageous in two ways; firstly, the markers can be fabricated using a process specifically tailored to the requirements of the markers in their role of guiding alignment and can be formed in materials which are designed to be ideal for markers. Secondly, it means that the device layers can all be aligned to the same reference point, and that drift during the writing of the first part of the device and the markers does not add to the error budget. When the markers are written at the same time as the first device structures, stage drift means that the relative positions are distorted by some unknown amount. When the second layer of the device is written it is aligned to the markers and the second device pattern is written in the expected position, however even with perfect alignment these two exposures will not precisely match. If the markers are exposed during a separate lithographic step and the first and second device layers are both written aligned to these

markers then any distortion in the marker's pattern will be the same for both layers of the device and hence the two device layers will exactly match. Any misalignment will be caused by errors in the alignment processes and not by uncontrollable stage drift during the writing of the markers.

The Penrose markers used throughout this project were fabricated using standard lift-off processes which will be described in section 4.3.1. Various different metals were used to create the markers however, it was finally decided that gold should be used because of its high atomic number contrast with silicon. Titanium was used as an underlayer to improve the adhesion.

Three considerations were taken into account when laying out the Penrose markers on a substrate. Firstly, as has been mentioned earlier in section 3.2.4, Penrose markers are only good at refining an already quite well aligned substrate, this is because a limited area of the marker is interrogated and the correlation only produces meaningful results when this region overlaps the region contained in the reference image by at least 50%. Therefore, conventional markers were included along with the Penrose markers. This is comparable with the use of distinct global and cell alignment markers, where the global markers are generally larger, more widely spaced markers that are easier to find when there is a large uncertainty in position. The cell markers are smaller, more precisely defined markers situated close to the device for best accuracy, and may be realised using Penrose markers.

Secondly, due to the mathematics of the projective mapping transformation used, the optimum layout of the markers is in the corners of a square. This maximises the sensitivity of the markers to the four classes of distortion to be corrected for. The markers can be placed in a different layout, however, care must be taken to ensure that all degrees of freedom are still measured. For instance, as explained in section 2.2 and figure 2.15 on page 41, if the markers are positioned in the centre of the edges of a square then they provide no measurement of the keystone distortion.

The third consideration relates to the length of time required to expose the pattern and the distance the stage must move between the marker searches and the exposure. The stage in an ebeam system is continuously drifting. This drift, in the worst case, is of the order of 10 nm per minute for the VB6, and means that if the exposure of a pattern lasts for a significant time then this drift will manifest itself as a positional error in the exposed pattern. This drift can be overcome if the alignment process is performed frequently enough or if the critical parts of the pattern are written relatively quickly after an alignment. Similarly, it is important that the markers and the area to be exposed are relatively close to one another because the distortions that alignment corrects for may vary significantly across small distances on the substrate's surface. The closer the markers are to the area to be patterned, the better since the corrections are defined as a differential and so any errors in the correction coefficients are magnified by the distance moved from the point at which the corrections were calculated.

A typical layout of markers on a wafer could be that shown in figure 4.9, where the squares are large conventional markers and the octagons represent the Penrose markers. The area to be written in each cell is indicated by the dashed outlines.

4.3 Experiments

Several experiments have been performed throughout the development of the correlation alignment process and in this section the details of each experiment and the results from it will be discussed. The experiments focussed on three areas; determination of the optimal process and imaging parameters, validation of the ability and accuracy of correlation to locate Penrose markers and lastly, measurement of the performance of the correlation based alignment process.

4.3.1 Marker Fabrication

Firstly, the ease of fabrication of different types of markers was investigated to allow an informed choice to be made based on marker reproducibility and process compatibility.

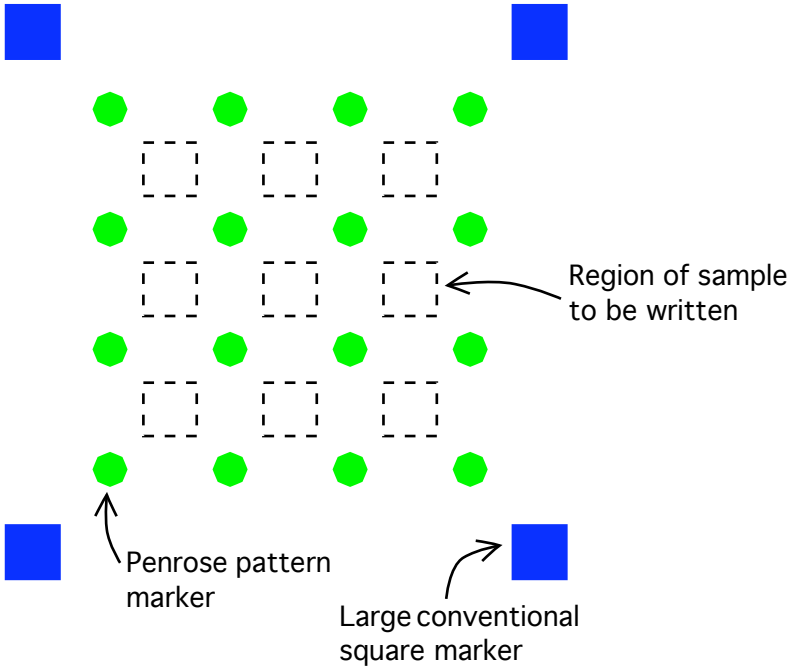


Figure 4.9: Typical layout of markers on a sample showing the large conventional markers for global alignment, with Penrose markers positioned in the corners of each cell, close to the area to be written and such that their sensitivity to each component of distortion is maximised.

This also allowed an investigation into the errors introduced during fabrication to be examined by using images of fabricated markers in simulations. The fabrication revealed issues when attempting to lift-off small areas enclosed on all four sides. This is a particular concern for the pseudo-random noise arrays, which contain many small, isolated squares, and supported the decision to use Penrose patterns. The simulations performed with imaged Penrose marker patterns showed that the sharp edges of the Penrose tiles can be replicated with sufficient accuracy to allow correlation to be performed and simulated shifts in marker position could be identified.

4.3.2 Optimisation of Image Capture Settings

Experiments were undertaken to determine the properties of the image collection system on the VB6 and the control that is afforded over settings such as the speed of image collection, the noise reduction filtering and the sample and frame averaging. For the alignment process to be a viable method of alignment there are certain requirements on each of these settings. The time taken to capture an image must not be prohibitively large relative to the improvement of the accuracy, otherwise the machine spends too great a proportion of time performing alignment procedures. The exposure then becomes an insignificant fraction of the job time. It is also important that the image collected is a true representation of the region of the sample interrogated. The use of RC noise filtering inevitable causes a delay to be introduced which would result in a systematic error in the located position and is therefore avoided. Lastly, it is important to be able to control the dose of electrons to which the resist on top of the alignment marker is exposed. Correlations performed off-line with the collected images confirmed that a compromise could be achieved between sufficient noise reduction and moderate interrogation time, while minimising the delay-induced image distortions.

The values used for the imaging in the subsequent experiments were as follows:

1. Size of image - 20 μm x 20 μm.

2. Number of pixels - 100 x 100 pixels.

3. The filter time constant - The minimum available filter time constant of 0 μs was used which effectively means that the pixel value was sampled immediately after the detector settling time.

4. Number of point samples - A single sample was taken at each pixel.

5. Number of full frame samples - A single frame was collected.

These values were used throughout the following experiments regardless of beam current or accelerating voltage used. The VB6 was always operated at 100 kV, whereas the experiments performed on the EBPG5000+ were performed at 50 kV. With these settings an image could be collected in approximately 8 s, and the correlation could be performed in approximately 600 ms, this is slower than the conventional marker search routine which can calculate the position of a marker in approximately 1 s.

There are similar parameters which can be applied to the conventional mark locate algorithm. For the following experiments a typical set of parameters were used with the mark locate algorithm to mimic the performance that could be expected in routine marker searches. No attempt was made to optimise these parameters for standard markers. The appropriate parameters are described below along with the values used throughout the following experiments.

1. Expected marker size, tolerance and the measurement length were set appropriately for the marker being interrogated. e.g. for a 10 μm octagonal marker, the tolerance was set to 10% and the measurement lengths were set to 0.5 μm.

2. The filter time constant - 16 μs was used as a means of noise reduction.

3. Number of point samples - 8 samples was taken at each point along each line scan again to reduce noise.

4. Number of scans per line - a single scan was performed along each line scan.

5. Parallel scans - 2 scans were performed on each edge of the marker to average out any variation in the marker's edge position.

6. MH & MW - the length of the marker's edge over which the parallel scan were performed. For 10 μm octagons this was set to 2.5 μm.

7. Rise time - the maximum distance over which the marker edge is expected to slope. Set to 0.5 μm to tolerate less well defined marker edges.

8. Contrast - the minimum change in video level that is treated as a marker edge. Set to 10% to allow thin metal markers to be located.

9. Coarse search limit - maximum area interrogated in attempt to find marker. This was set to 100 μm for a 10 μm octagonal marker.

4.3.3 Validation of Correlation

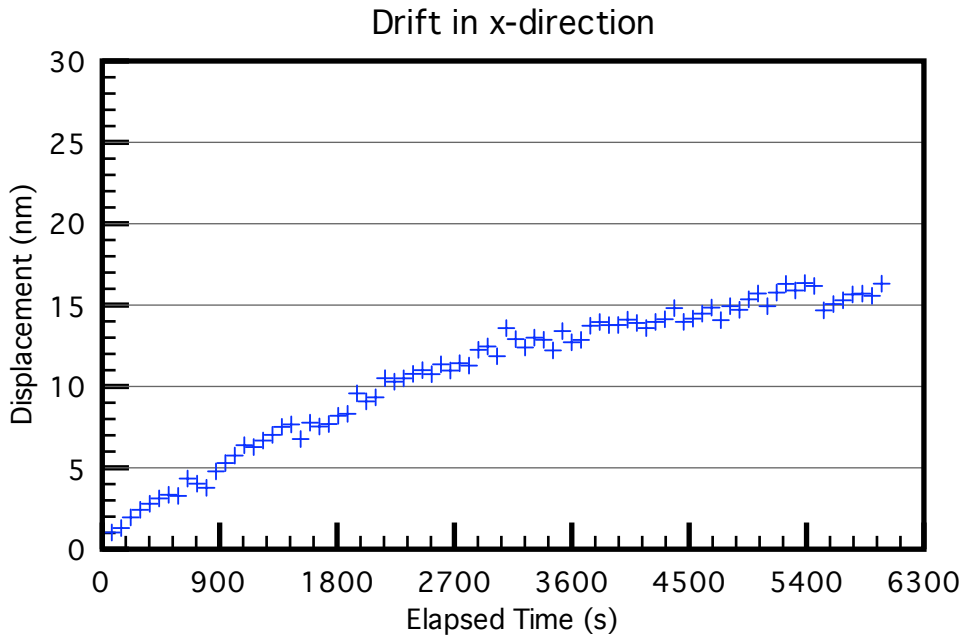
Before a full alignment was attempted an experimental validation of the ability of a correlation process to locate Penrose markers was performed. To investigate the accuracy with which markers could be located using correlation the normally undesirable attribute of stage drift within the electron beam system was used to allow the effects of small displacements to be investigated. Stage drift is an inherent problem with electron beam systems which causes slight shifts in the position of the stage over time. The shifts are the result of varying conditions such as temperature or electromagnetic field within the electron beam lithography tool. There is always some drift due to electronics such as from the HT power supply or scan generator. Secondly, although there are strict environmental controls, the room temperature will vary slightly and this also introduces drift. Whenever a holder is loaded onto the stage there is usually a slight difference in temperature and this introduces drift as the holder temperature relaxes to that of the stage. Therefore, it is expected that after mounting a holder on to the stage the rate of

the drift should decrease with time as the system comes closer to the background drift rate. This background drift rate has been measured previously to be around about 6 nm per hour.

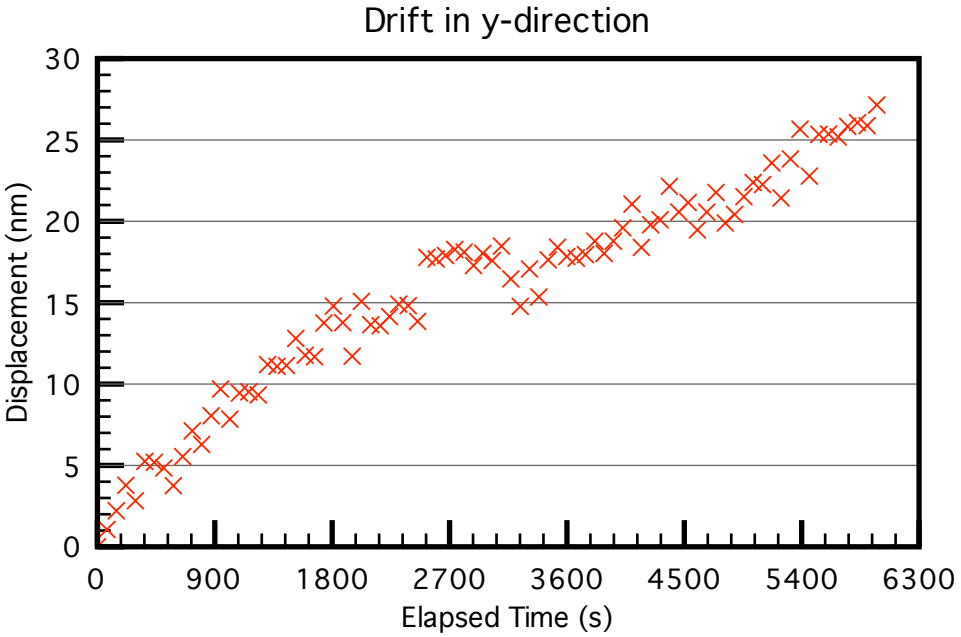
To measure the ability of correlation to measure stage drift and hence to measure small displacements of Penrose marker the stage was positioned at a Penrose marker site. Images of the Penrose pattern were regularly captured over a period of about two hours starting shortly after the holder was loaded onto the stage. As each image was obtained, the correlation program produced the correlation of that image with the first image taken, and from that calculated the relative displacements in the x and y directions caused by the stage drift. The results are shown in figure 4.10(a) and 4.10(b) for the x and y directions respectively.

Two interesting observations can be made from these results; firstly, the general shape of the curves fits with the expected decrease in the rate of drift as time increases. This is particularly clear from the x-offset graph but more data is perhaps needed to confirm the trend for the y-offset curve. The two curves both fit in with the magnitude of the expected background drift rate which during installation of the tool was measured to be of the order of 6 nm per hour after the stage was allowed to thermalise overnight. Both trends however, tend to show a larger value of drift which can be explained by the fact that the experiment was carried out directly after loading the holder onto the stage.

The second observation is the apparent discrepancy between the noise levels between the two directions, with the x-direction displaying an apparently less noisy curve. To investigate this further the standard deviations of the points from a least squares fitted 5^{th} order polynomial were calculated to be around 0.5 nm in the x-direction and 1.1 nm in the y-direction. (A 5^{th} order polynomial was used as it gave a good approximation to the general trend of the data without smoothing out random errors between measurements.) This gives an initial estimation of the random error associated with using a correlation method to locate a Penrose marker and, while not a direct measure of such errors,



(a) Displacement of Penrose marker in the x-direction showing a measurement of the stage drift over time.



(b) Displacement of Penrose marker in the y-direction showing a measurement of the stage drift over time.

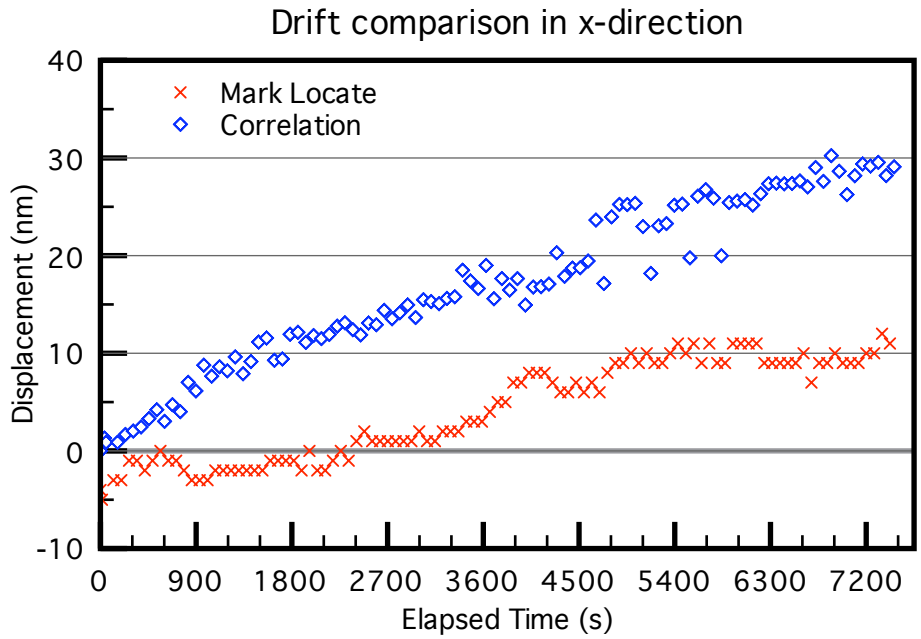
Figure 4.10: Displacement of a Penrose marker pattern as calculated by the correlation program giving a measure of the stage drift over time.

are encouraging initial measures of the possible alignment accuracy attainable with the correlation based method.

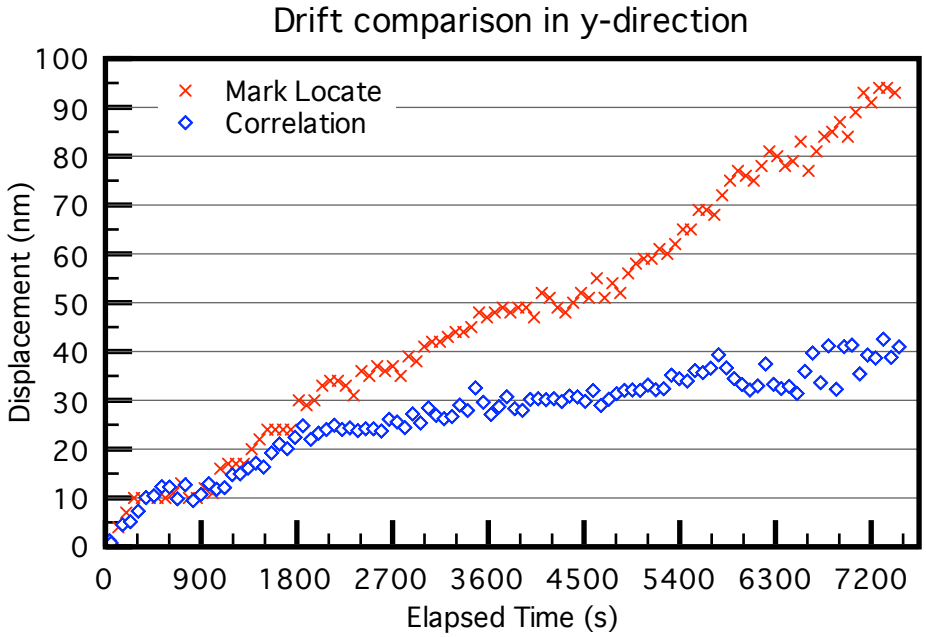
4.3.4 Correlation Performance Comparison

To gain an insight into how correlation based alignment with Penrose patterns compares to the conventional alignment 'mark locate' routines a measurement of stage drift by both correlation with Penrose marker and using conventional mark locates was performed. A set-up similar to that of the above experiment was used, except that both a Penrose marker and an octagonal marker were interrogated during each iteration. A program collected an image of the Penrose marker from which a measure of the offset was obtained using correlation. Immediately after collecting the image a conventional mark locate was performed on an octagonal marker, situated on the holder marker block, to see how the drift would have been measured conventionally. Again the measurements were taken over a period of approximately 2 hours starting immediately after the holder was mounted onto the stage. The graphs of figure 4.11 show the measured values for the offsets in the x and y directions from the two methods.

Although the magnitudes of the offsets as measured by the two different methods are not in good agreement, particularly in the y-direction, there is however a general agreement on the shape of the curves. The difference in measured drift between the two methods is consistent with a differential thermal expansion. The difference in locations between the Penrose marker on the sample and the traditional alignment marker located at the edge of the holder on the holder marker block was estimated to be about 20 mm in x and 50 mm in y. The measured difference in drift of 50 nm in y corresponds to a differential motion of 1 ppm of the separation. Since the coefficient of thermal expansion of the Aluminium stage is 23.1 ppm K^{-1} this would correspond to an overall change in the stage temperature on loading of 0.043 K consistent with the specified room temperature accuracy of ± 0.1 K. The measurements in x are closer since the difference in positions of the two markers is also smaller in x.



(a) Comparison between the drift in the x-direction measured by the conventional mark locate and the correlation processes.



(b) Comparison between the drift in the y-direction measured by the conventional mark locate and the correlation processes.

Figure 4.11: Comparison between the drift in the x and y directions as measured by the conventional mark locate and the correlation processes.

In order to further compare the accuracy of the conventional mark locate procedure with the correlation based method on Penrose patterns a substrate was prepared with octagonal and Penrose markers situated just 20 μm apart. A pattern consisting of two 10 μm octagons and two 8 μm Penrose patterns forming a square, as in figure 4.12, was arrayed several hundred times and exposed on a Si wafer. The markers were fabricated using lift-off of 10/60 nm Ti/Au.

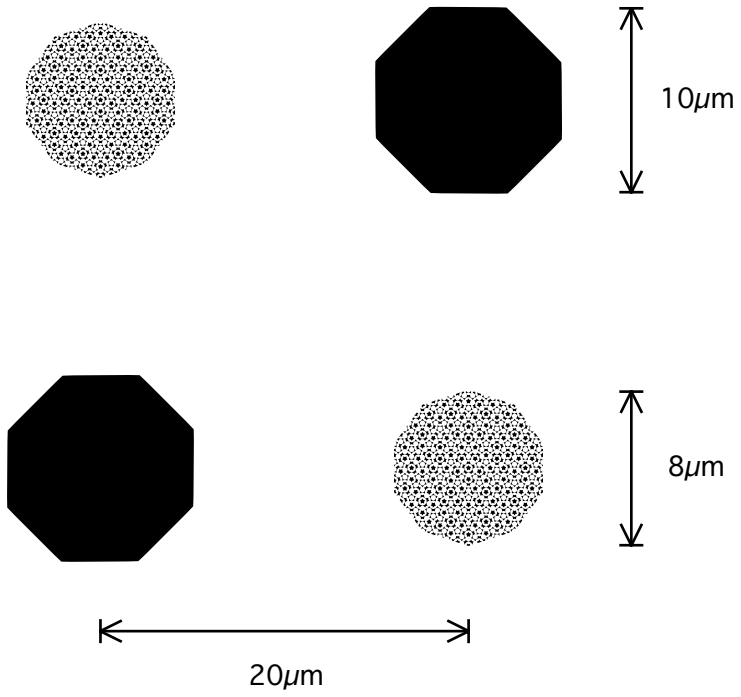


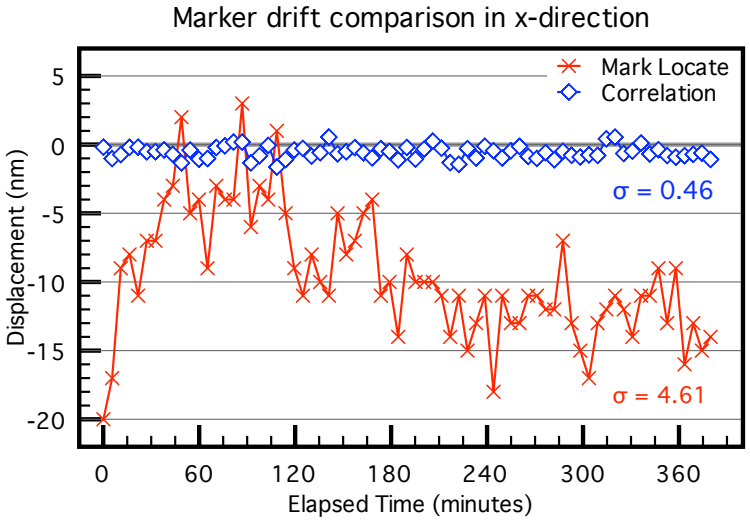
Figure 4.12: Layout of Penrose and octagonal markers used to compare the accuracy of the correlation based marker locates with that of the conventional mark locate algorithm.

To compare the performance of the two methods of locating markers, the positions of three adjacent markers were repeatedly measured. First one Penrose marker was imaged and its offsets computed by correlation. This measurement was used as the origin for the other two offset measurements such that the effect of any stage drift could be removed from the offset measurements. One of the octagonal markers and the other Penrose marker were located and the offsets of these markers calculated. In this way effectively the

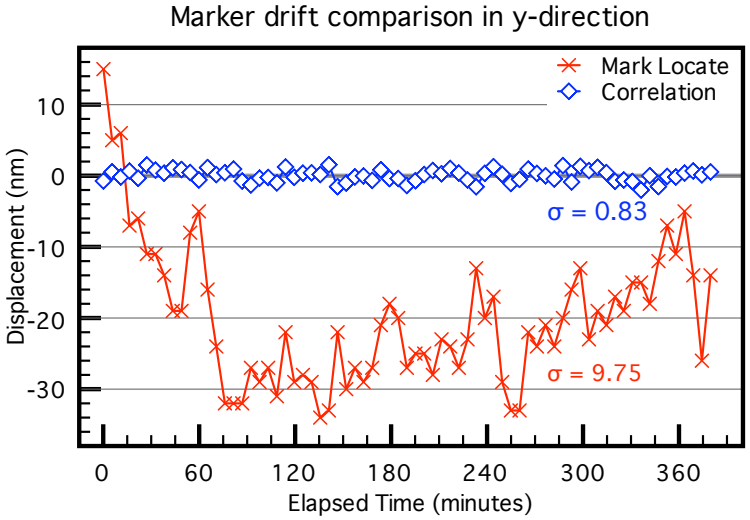
separation between the first and second Penrose marker and the separation between the first Penrose marker and the octagonal marker were repeatedly measured and calculated. This interrogation of the three markers was repeated over 6 hours, starting shortly after the holder was mounted onto the stage, to give a measure of the accuracy of the two mark location methods. Figure 4.13 shows the measured offsets as a function of time.

The difference between the curves for the correlation and the mark locate methods, may be explained in several ways. Firstly although the stage drift has been removed by using one of the Penrose marker measurements as the origin any differential motion between the markers would show as an offset on the graph so for instance the sample will expand or contract with temperature changes and so the separation of the markers will correspondingly vary. The markers are however separated by only $20\,\mu$m and the coefficient of thermal expansion of Si is $2.6\,\mathrm{ppm\,K^{-1}}$ this would not account for all of the measured offsets. However any rotation of the substrate caused by the thermal expansion could account for the opposite trends in the mark locate curves in the x and the y-axes. Measuring a Penrose marker with correlation and using it as a measure of the origin means that all the points in figure 4.13 will have the measurement error associated with the correlation based method, therefore the σ values will incorporate this as well as the errors in the second correlation or the mark locate method. Thirdly, it may also indicate that the correlation algorithm cannot measure as small displacements as the mark locate algorithm however the previous results show that correlation is capable of measuring displacements of this magnitude so this can be ruled out.

The σ values quoted on the graphs of figure 4.13 give the standard deviation of each of the curves and since the curves only vary with errors in the alignment algorithm this gives a measure of the attainable accuracy of the correlation and mark locate routines. There is a marked difference between the x and the y-directions, with the values in the y-direction generally being worse. This can most likely be attributed to the presence of approximately 3 nm of vibration on the beam in this direction. In comparison with the



(a) Comparison between the drift in the x-direction measured by the conventional mark locate and the correlation processes.



(b) Comparison between the drift in the y-direction measured by the conventional mark locate and the correlation processes.

Figure 4.13: Comparison between the drift in the x and y directions as measured by the conventional mark locate and the correlation processes.

mark locate results the correlation algorithm produces around ten times better accuracy, 0.46 nm as opposed to 4.6 nm in the x-direction, and 0.83 nm as opposed to 9.8 nm in the y-direction. These figures are extremely promising and suggest that sub-nm alignment is possible using a correlation based method.

4.3.5 Metrology

Measuring the performance of an alignment process is a significant problem in itself. There are several factors that must be considered to ensure that a meaningful measurement of alignment is obtained which is also robust against some of the variabilites of any fabrication process. A straightforward method to measure alignment is to create a ruler in the first process step, this can be concurrently with the markers or as the first aligned exposure, then to expose a tick mark next to the ruler at the zero point in a subsequent process step. Any misalignment between the writing of the ruler and the writing of the tick mark will result in the tick mark being displaced from the zero point on the ruler, for large enough displacements the misalignment can be read off the ruler, however it becomes very difficult to accurately measure alignments of the order of 10 nm with this method.

Perhaps the most widely used and accepted method of measuring alignment is by fabricating a pair of rulers, side by side, with slightly differing periods so as to form a Vernier scale [119], such as in figure 4.14. Like the scale on Vernier callipers this magnifies small distances by using the beat frequency of two gratings with slightly different periods. For example if in the top ruler the marks are spaced 100 nm apart but the tick marks are 105 nm apart in the bottom ruler, there is a 100:5 magnification of any displacement. A displacement of 5 nm will mean that only the first tick mark (at a distance of 100 nm from the centre of the Vernier) on each grating will exactly align to one another. Thus by finding the tick marks that most closely match a measure of the misalignment can be made.

These methods both rely on using the edges of the exposed lines to define their

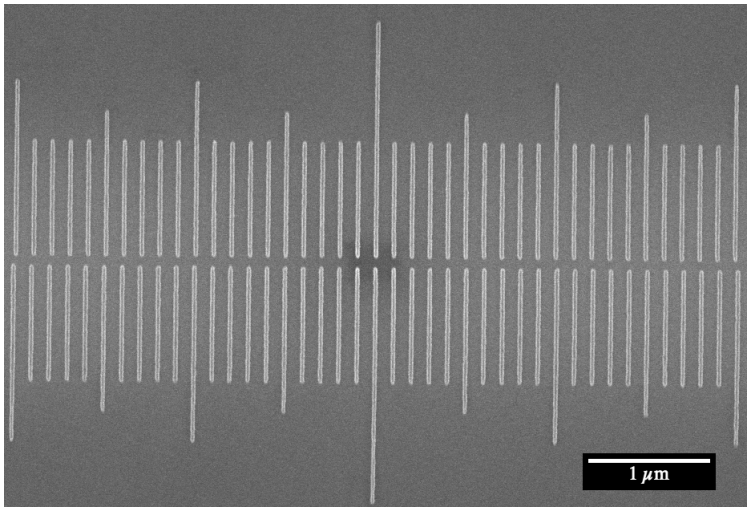


Figure 4.14: Vernier structure which can be used to measure small displacements between the two halves of the Vernier, and as such can be used to measure alignment.

position; the problem with this is that the line edges are dependant on exposure dose, development conditions, resist thickness and the pattern transfer process as well as the alignment accuracy. However the period of the gratings is well defined and also unaffected by the dose, development, resist conditions or pattern transfer process, only being affected by scale distortions during exposure.

To take advantage of this property of gratings a method of measuring alignment using the period of defined structures has been devised. It relies on forming a single grating but writing alternate lines of the grating in two different exposures. The first half of the grating with a period $\frac{2}{f}$ is written during the first exposure. The second half of the grating also with period $\frac{2}{f}$ is then written offset by $\frac{1}{f}$ from the first half in a second exposure. The resulting structure is an overall grating with period $\frac{1}{f}$. To analyse the fabricated grating a SEM image is taken and then Fourier analysis is used to allow the misalignment between the two exposures to be determined. When the two halves of the grating are perfectly aligned a Fourier analysis of an image of the complete grating shows that only the fundamental frequency, f, is present in the power spectrum. However, any

misalignment between the first and the second half of the grating results in a component at the half-frequency, $\frac{f}{2}$, as shown in figure 4.15. Furthermore, the energy in the $\frac{f}{2}$ component is directly proportional to the energy in the difference and so the amplitude of the $\frac{f}{2}$ component is directly proportional to the misalignment between the two halves of the grating. Therefore it is possible to get a measure of the misalignment by comparing the power contained in these two peaks. The reliance on the differential between two measurements in the frequency spectrum has the added advantage of removing any error caused by the limit in the absolute accuracy of the SEM which can be estimated to be around 5% [120].

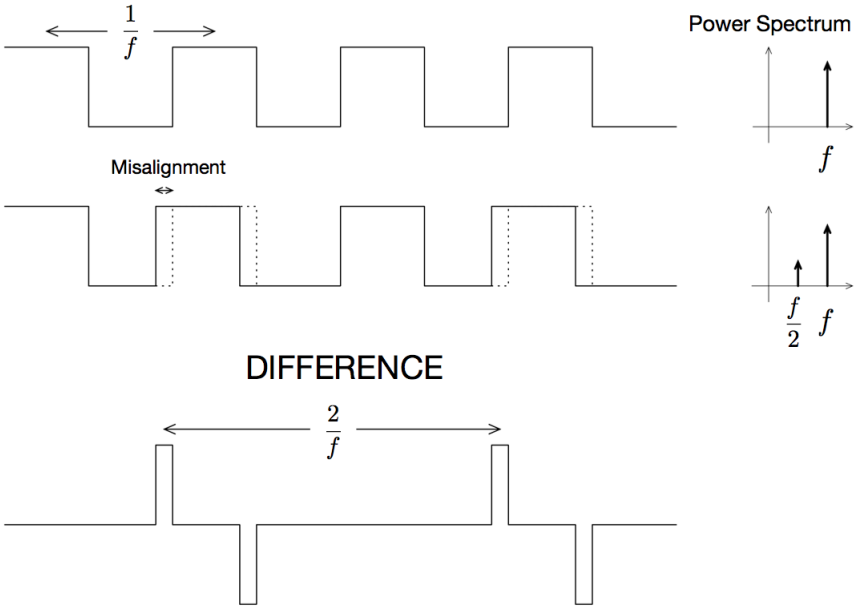


Figure 4.15: Fourier analysis of an overlaid grating showing that the amplitude of the $\frac{f}{2}$ component in the power spectrum is directly proportional to the misalignment between the two halves of the grating.

Exposing several of these gratings with a range of deliberate misalignments e.g. from -10 to 10 nm in 1 nm steps, analysing each grating, and plotting the magnitude of the $\frac{f}{2}$ peak against the induced offset gives a direct measure of the offset with the minimum alignment error. This 'V'-shaped curve can be used to give an accurate measure of the

misalignment between the two halves of the grating.

Gratings with a period of 50 nm have been written in HSQ in two stages as described above and imaged in an Hitachi S4700 SEM, figure 4.16. The Fourier transform for each of the gratings in figure 4.16 has been taken, an example power spectrum from one of the misaligned gratings is shown in figure 4.17, where the f and $\frac{f}{2}$ peaks can be clearly seen.

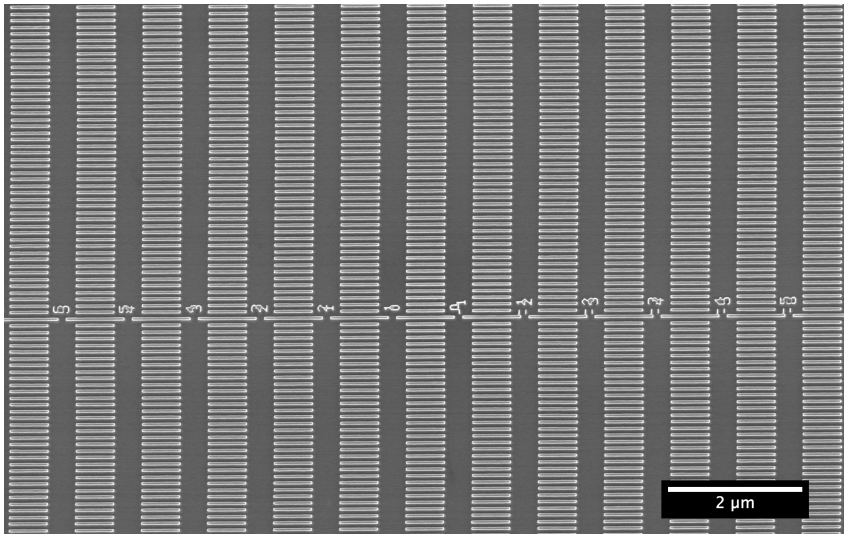


Figure 4.16: SEM image of overlaid gratings with period of 50nm and offsets from -6 to 6nm.

This analysis is repeated for each grating and the resulting plot of half-frequency amplitude against positional offset is show in figure 4.18. A least-squares fitting has been performed on this data to a 'V'-shaped model of two straight lines with the same slope and from their intercept a measure of the misalignment can be obtained. The results give a measurement of -0.63nm for the alignment. The errors in the magnitudes of the $\frac{f}{2}$ peaks in figure 4.18 most likely arises from the measurement error in the SEM as any scale distortion across the SEM scan field would produce slight changes in the measured period as would any uneven variation in the flaring across the SEM image.

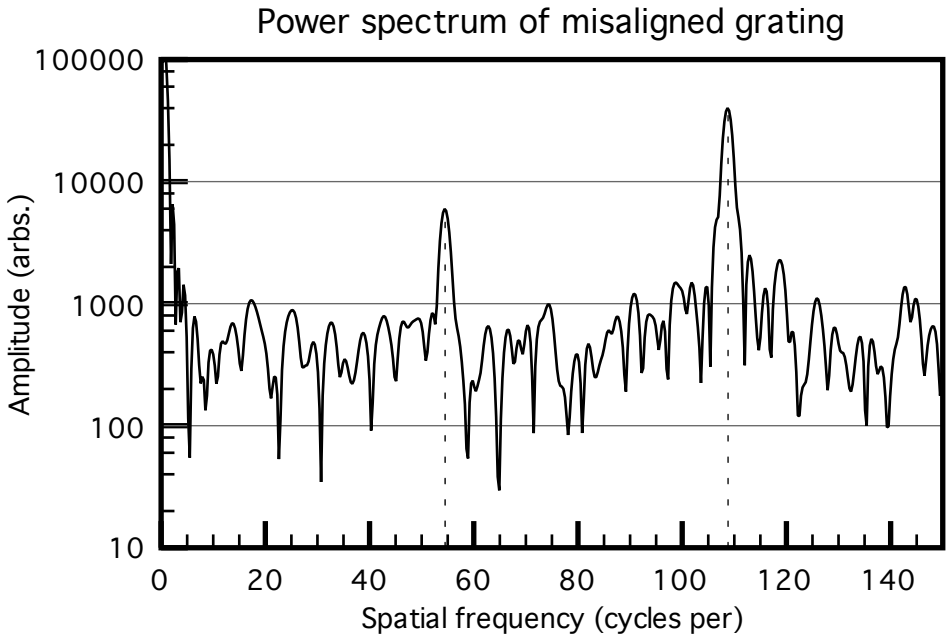


Figure 4.17: Example power spectrum for one of the misaligned gratings shown in figure 4.16.

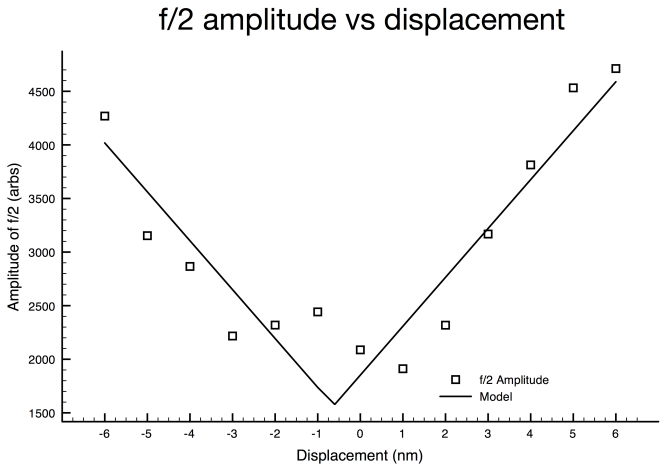


Figure 4.18: Graph showing the amplitude of the $\frac{f}{2}$ component of the power spectrum plotted against offset, with 'V'-model fitted demonstrating alignment measured at -0.63nm.

5 Robustness to Marker Damage

Markers can be subject to various types of damage and it is obviously advantageous for alignment techniques to be able to align to damaged markers. One of the potential disadvantages to using Penrose patterns as alignment marks is the complexity of fabrication. Since the marker ideally consists of many elements fabricated near to the resolution limit of the fabrication process there is a greater susceptibility for any variations during the fabrication process to cause substantial errors in the pattern transferred to the substrate. It is not immediately clear how errors in the reproduction of Penrose pattern markers affect the alignment accuracy. However, with traditional ebeam alignment using uniformly filled polygonal markers, there is a high susceptibility to edge defects. This includes things such as "rat bites", edge roughness and "flagging". In correlation-based alignment of Penrose pattern markers, there is less susceptibility to these edge effects since many more edges contribute to determining the marker position, and as the disturbances in the pattern are random, the larger total edge length involved in determining the marker's position helps to average out any errors. Since markers are generally large, thick structures it is common, in III-V semiconductor processing, to create them at the same time as Ohmic contacts. Ohmic contacts are annealed during fabrication, to help the alloying of the metal layers and to improve the contact resistance, and any marker created at the same time will also be subjected to the high annealing temperatures. The temperature used during annealing allows the metal to recrystalise, tending to equilibriate the stress within the metal layers and therefore rounds sharp corners and edges into

more smoothly flowing structures. This obviously changes the defined shape and profile of the markers and can leave the markers looking considerably different to their designed shape.

There are also fabrication defects associated with creating smaller features such as those used in Penrose marker patterns. Several types of defect are discussed in depth below and, after the descriptions, their effects on the mark location process are analysed.

5.1 Types of Damage

5.1.1 Edge Defects

There are several types of edge defects that can affect markers. The main types of edge defects are "bias", "roughness", "flagging" and so called "rat-bites" where part of the edge has been removed.

A *bias* is where the edge itself is well defined but the position at which it is created is different from the designed position. This could arise due to poor collimation of the metal atoms being evaporated, resulting in the position and size of the marker being dependant on the distance and angle between the point on the substrate and the crucible, especially with thick resist.

Edge roughness is where, rather than being perfectly smooth, an edge has some variation in position about the optimal line. Several factors may contribute to the edge roughness but it is substantially due to noise in the pattern definition and transfer processes used to create the marker. A contributory factor might be the resolution of the resist which results in the lithographically defined pattern not being accurately reflected in the resist layer. It could also be caused by slight variations in the chemical composition of the resist, local fluctuations in solvent concentration during development which on a microscopic scale produce slight variations in edge position, or statistical variation in local exposure dose due to shot noise in the beam.

For electron beam lithography the proximity effect [114] can have a significant effect

on the edge position particularly when the markers are not surrounded by symmetrical features. This is perhaps most noticeable at the corners of pattern which end up more rounded with internal corners tending to fill-in and external corners tending to be under-filled, but it affects straight edges where there are changes in the pattern density locally (within a 10 μm range but this values depends of the accelerating potential and substrate material used).

These effects arising due to defects in the resist pattern cause problems for the definition of the marker whatever pattern transfer process is used to create the marker. In addition to defects in the resist, the pattern transfer process itself can also introduce edge roughness.

For a lift-off type process evaporation of the metal can produce slight variations in the edge position due to the grain size of the metal being evaporated, and the tendency of the metal to change shape to minimise grain boundary energy. This limits the smoothness of the edge defined in the metal as shown in figure 5.1, which shows an SEM image of a thick aluminium layer with large grains after it has been wet-etched, leaving poorly defined edges.

Furthermore, the resist profile used for lift-off means that the top layer of resist acts as a shadow mask blocking the evaporated metal from certain parts of the underlying substrate. Over time metal grains can form on this edge and therefore change the profile of the metal that reaches the substrate. This too can cause irregularities in the edge definition as it results in variations in the thickness of the metal at the edge. Furthermore, the varying strength of bonds between the metal and the substrate material mean that some metals adhere more strongly to the substrate than others. In places where there is poor metal adhesion to the substrate a lift-off process can remove metal that was intended to remain on the substrate and this can cause defects in the metal pattern.

When an etching process is used to create the marker the edge roughness effects come from slightly different sources. Principally they arise due to local variations in the

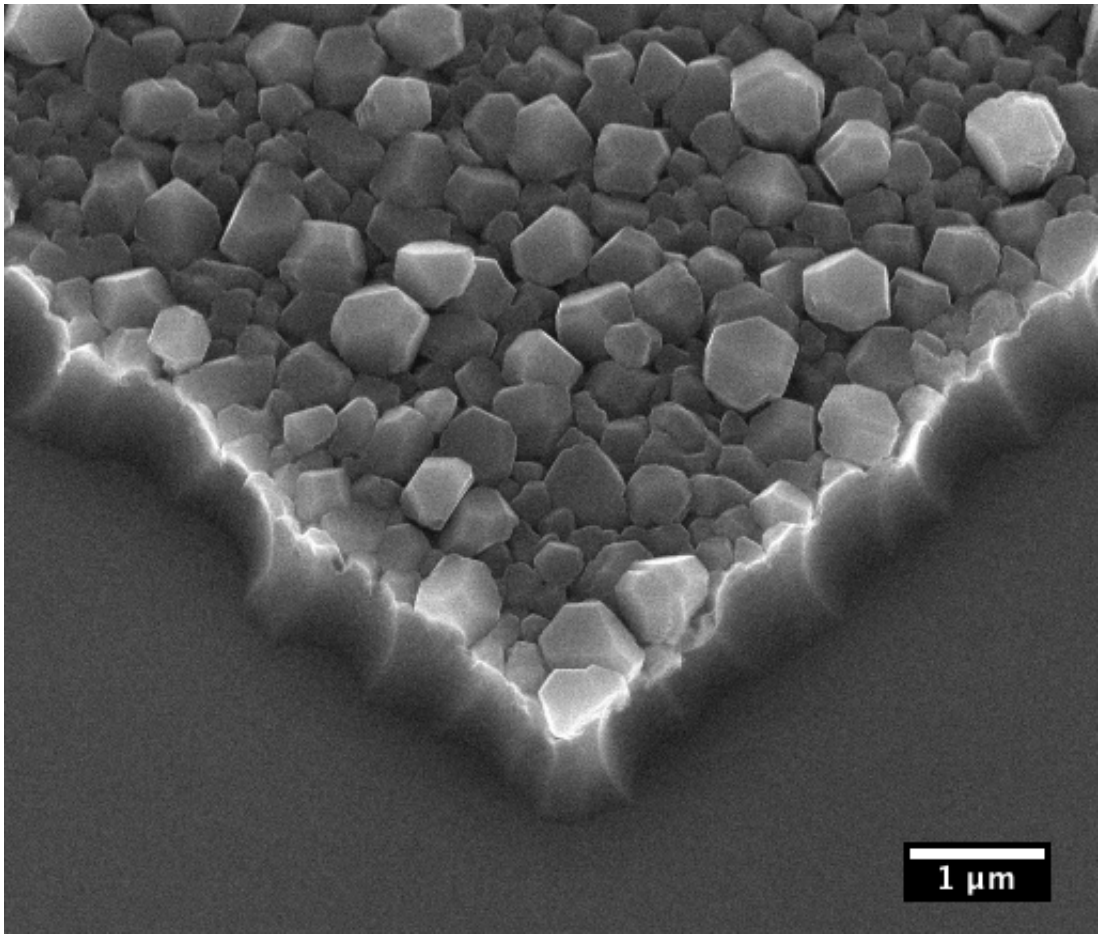
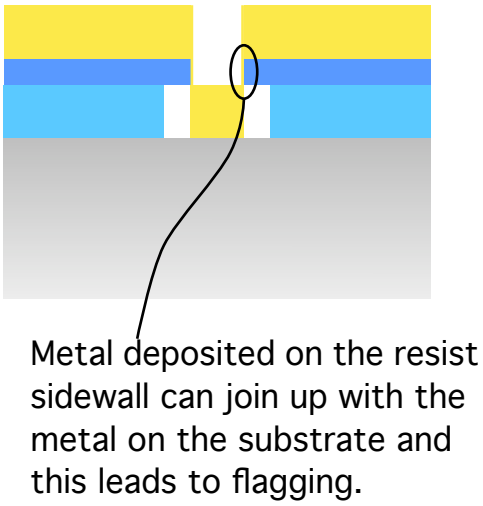


Figure 5.1: SEM image of a thick Al layer with large grains after wet-etching showing the large local variations in the edge position due to the grain size of the metal. *Courtsey of D. McCloy, Kelvin Nanotechnology Ltd.*

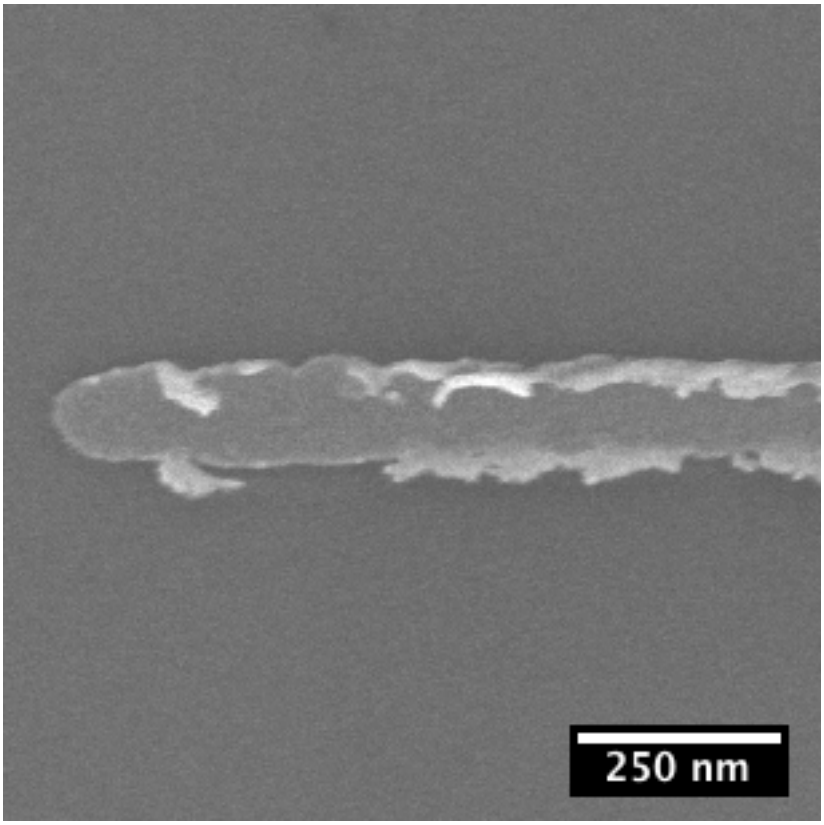
chemical reaction on a microscopic scale, especially anisotropy of etching with respect to crystallographic orientation, which produces slight variations in the local etch rate. Things like small resist residues on the metal surface can reduce the local etch rate and so leave slightly more metal at these points. Conversely a surface which is pitted can have a higher local etch rate, resulting in less metal remaining at these points. Variations in "induction time", the time before the metal begins to etch, in a dry-etch process can be caused due to surface layers on the metal, especially oxides. The result is similar to those for wet-etching. For dry-etch processes that use physical etching there can similarly be local variations in the etch rate, due to micromasking, leading to roughness in the defined edge. These local etch rate variations produce local positional variations in the marker's edges or in the local thickness of the deposited metal and reduce the smoothness of the edges.

Flagging happens only during lift-off processes and occurs where there has been an inadequate separation between the top of the metal layer and the bottom of the top layer of resist, a situation shown schematically in figure 5.2(a). This results in a vertical build up of metal at the edge of the defined regions which remains after the resist has been removed, as shown in the SEM of a metal line with flagging in figure 5.2(b). The remaining vertical strip of metal is known as a "flag". The flag does not necessarily have to be standing vertically and it can be quite short in length. These pieces of additional metal at the edge of the marker can overhang the edge and change its profile, as shown.

While flagging is a defect that results from additional pieces of metal around the edges of the marker, rat-bites are caused by the removal of certain parts of the metal at the edge of a marker. There are several possible reasons for rat-bites to be formed; for example particles or resist residues can cause poor adhesion or masking of the metal at a specific point on the substrate. Poor adhesion of the resist can allow etchant to seep under the resist and allow it to attack parts of the metal that would become the marker. Stress in the deposited film can also cause fissures to form and parts of the metal marker



(a) schematic showing resist profile leading to flagging



(b) SEM image of a line of evaporated metal with visible flags

Figure 5.2: Illustration of the physical process that produces flagging and a SEM image of a metal wire with visible flagging.

to become detached from the substrate leaving parts of the marker missing. Also during lift-off, if the metal is not well adhered to the substrate, then parts of the metal film can rip parts of the marker metal away leaving notches in or missing parts of the marker. Figure 5.3 shows BSE image of a square marker with a section of the metal missing from the lower-left corner.

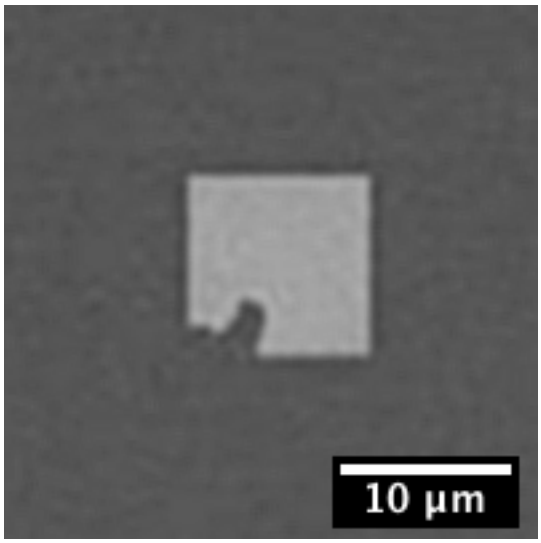


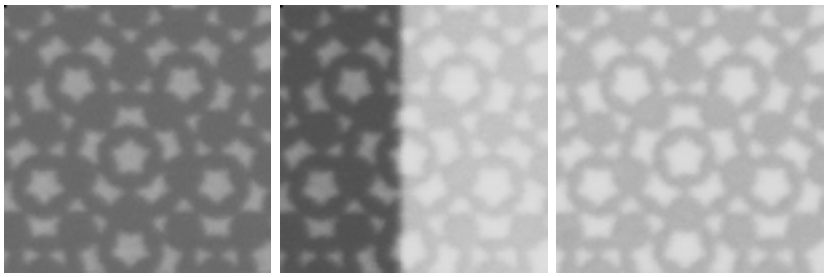
Figure 5.3: Square marker with a rat-bite in the lower-left corner.

Edge defects are most significant for conventional polygonal markers. This is due to the fact that relatively few edges are used when locating the marker so any defects with a single edge can profoundly affect the detected position. Using more complex marker patterns with an increased number of edges, along with a method that can recover and use this additional information, allows a tolerance of defects affecting a single edge meaning that they have a much less profound impact on the locating process. In this respect Penrose marker patterns are almost ideal because they are formed from many similarly sized elements, which are all relatively small, and exhibit $\sim 50\%$ fill, so they maximise the number of edges in a given area. However edge defects are only one class of marker defect and there are several other types of defect that can occur. Three classes of defect

are now examined and their effects on Penrose markers investigated.

5.1.2 Additional Metal Layers

Additional metal layers could be classified as additive defects (see section 5.1.3), however they are a special case, and since they are commonly encountered they merit additional discussion and so will be dealt with here separately. The discussion is restricted to cases where a uniform coating of metal covers a sizeable proportion of the surface area of the marker, such that this additional layer has no or only one edge within the marker area, i.e. it is a sheet laid on top of the marker, covering or partly covering the marker. They often occur when markers are reused and robustness to this type of damage has a large practical value. The metal layers left behind can vary in thickness and extent depending on the metal used in the lift-off process and on how well the metal has adhered to the region around the marker. Figure 5.4 shows a range of metal layers covering different proportions of a Penrose marker.



(a) no additional metal layer (b) partial additional metal layer (c) uniform additional metal layer

Figure 5.4: Penrose pattern markers with additional metal layers covering various amounts of the marker.

In the first instance, figure 5.4(a) there is no additional metal, the marker is unchanged and it is the ideal case and the alignment accuracy attainable will reflect this. At the other extreme case, figure 5.4(c), where there is a uniform film covering the entire marker, it is clear that although there is an increase in the BSE signal from the background areas there is minimal loss of positional information because all the marker edges remain

distinguishable. In Fourier space the metal over layer corresponds to a change in the DC component and so doesn't affect the positional information content of the image. Coupled with the fact that correlation is a linear process this means that the additional metal layer has no effect on the accuracy with which the position of the marker can be found.

When the covering layer is not complete as is often the case interesting things happen to the correlation process; a shift is induced in the found position. In the worst-case scenario where half the marker is covered, as in figure 5.4(b), the offset in the found position can be shown to be equal to half a pixel, which equates to 10nm in this case.

5.1.3 Additive Defects

Additive defects occur when material is deposited on top of a marker once it has been fabricated. This can occur for various reasons, for example, particulates, flagging during lift-off working free from the pattern edges and being redeposited on a different part of the pattern or additional material being left from another processing stage. When the deposited material completely obscures sections of the marker pattern there will be some loss of information. It is obvious that the percentage of the interrogated area that is occluded, will influence the extent of the effect on the mark locate accuracy, but this is difficult to quantify analytically. Instead the effect that additive defects have on the mark locate accuracy has been investigated experimentally by fabricating Penrose marker patterns with segments deliberately obscured and systematically varying the area covered by these defects. This is valid since the presence of dust or particulates on the pattern obscures the pattern data: adding a 'dust' pattern also obscures the pattern data but in a controllable fashion suitable for systematic investigation.

The markers were fabricated on blank Silicon wafers using a 150 nm thick bi-layer of PMMA using a dose of 700 μCcm^{-2} and a 1 nA beam current on the VB6 running at 100 kV. The patterns were transferred into metal by lift-off of 10 nm of Ti and 60 nm of Au. The marker patterns used were the standard Penrose marker patterns that have

had octagonal defects overlaid. These defects range in diameter, from 10 nm to 600 nm and in density from 1 to 10 defects per μm^2. Since these defects are part of the pattern file used to define the markers they are exposed at the same time and are lifted-off in the same layer of metal. Therefore they have the same contrast and completely replace that section of the marker, removing all positional information. Figure 5.5 shows a range of markers with a) varying density of defects and b) varying size of defects.

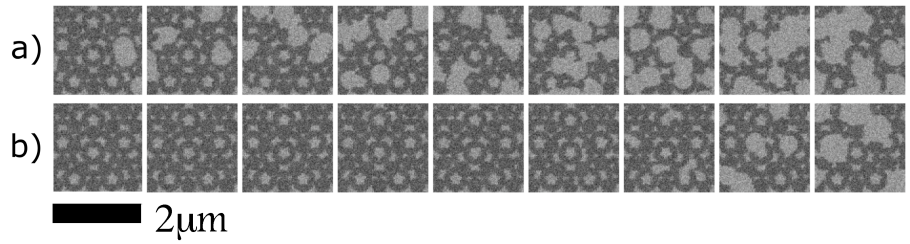


Figure 5.5: Examples of the fabricated Penrose markers with added defects ranging in a) density and b) size.

5.1.4 Subtractive Defects

The converse of additive defects where additional material covers part of the marker, are subtractive defects where part of the marker pattern itself has been removed. These can occur for several reasons; poor adhesion of the marker metal to the underlying substrate in a lift-off process, poor adhesion of the resist to the metal in an etching process, or from particulate contamination on the substrate masking the pattern in a small locality. Subtractive defects can also be caused by underexposure or underdevelopment resulting in the size of the pattern being reduced.

A second set of marker patterns were created in a 150 nm thick bi-layer of PMMA using a 1 nA beam current on the VB6 running at 100 kV. The patterns were transferred into metal by lift-off of 10 nm of Ti and 60 nm of Au. Sections of the pattern were deliberately removed and this was achieved by reducing the dose used during the ebeam exposure of the markers. This has two effects; as the dose decreased from the ideal

dose-to-size for the pattern elements, the size of the individual elements reduces. This also allows us to see the effect of many small changes in marker edge positions. Further reduction of the dose caused elements of the pattern to be omitted until at the lowest doses none of the marker pattern was transferred into the lifted-off metal. This range is shown in figure 5.6 and the dose used to expose each marker in the figure is given in table 5.1.

Marker Number	Dose (μCcm^{-2})
7	490
8	507
9	524
10	542
11	561
12	580
13	600
14	621
15	642

Table 5.1: Doses used to expose the markers of figure 5.6.

Each of the marker patterns was written by alignment to four good Penrose markers around the corners of the written area. This allowed them to be created at a known and well-defined position. To investigate the accuracy of the mark locate algorithm an additional alignment step was performed with the good Penrose markers, but rather than exposing a pattern an image was collected aligned to the exact centre of the damaged marker pattern. Thus if alignment to the damaged mark were to be perfect, the measured displacement would be zero. Correlating this collected image against the pre-defined reference image allows us to get a measure of the error in the marker search induced by the damage to the marker pattern. Repeating this process at several markers of each type of damage allows us to measure the variation in the marker searches and hence to assess the achievable alignment accuracy from the standard deviation of the marker searches.

The experimental layout is shown in figure 5.7. Schematically, the process was as

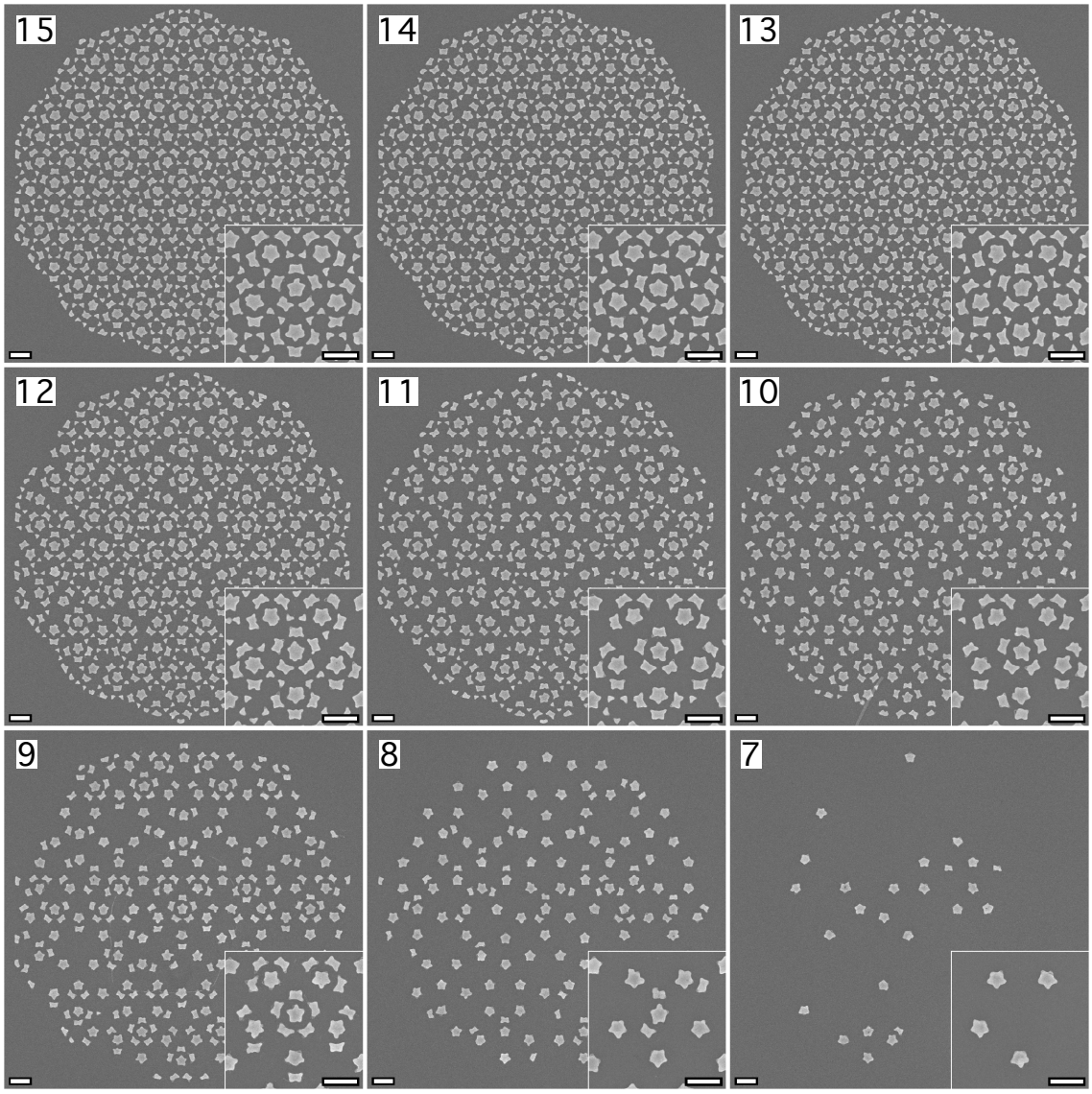


Figure 5.6: Examples of the Penrose markers fabricated by varying the exposure dose demonstrating the changing size, and the omission of pattern elements as the dose reduces. Each marker in the range was numbered corresponding to the dose used, as shown in table 5.1, and these will be referred to later. The scale bars shown are 500 nm long.

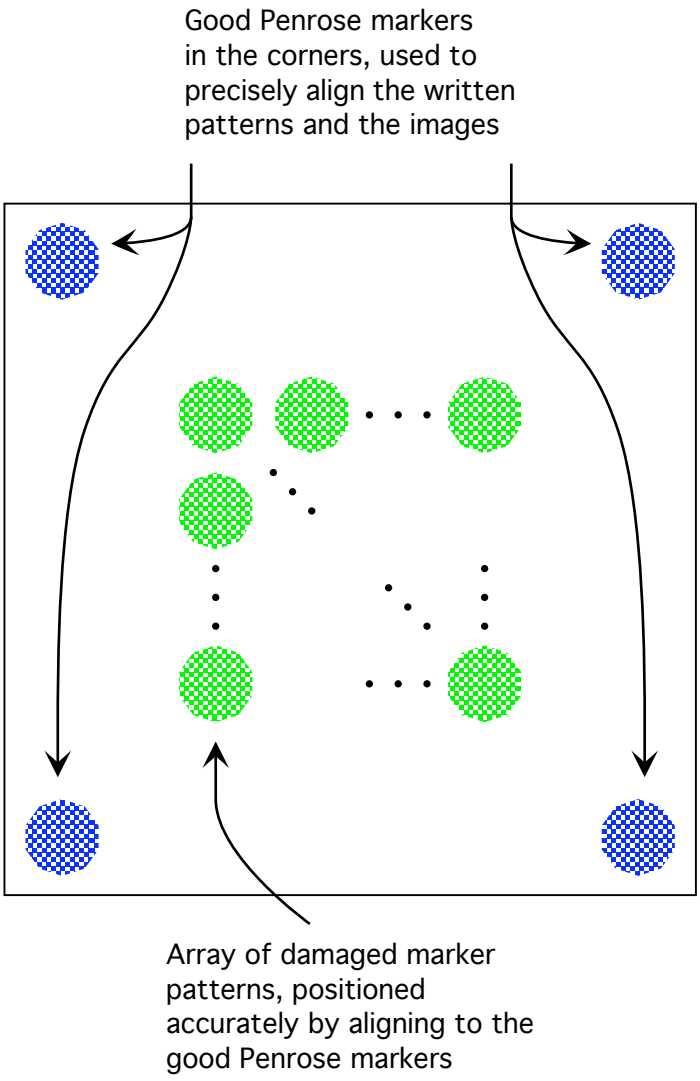


Figure 5.7: Schematic of the layout used to accurately expose and image the damaged markers. Showing the good Penrose markers in the corners which allow each of the damaged marker patterns in the array to be placed in known, well-defined positions and for the exact centre of these markers to be imaged.

follows.

1. Prepare the sample with good markers, lifted-off in 10/60 nm Ti/Au metal.
2. Spin the resist on the sample ready for the damaged marker patterns to be written
3. In the VB6 ebeam tool, align to the good markers and write a damaged marker pattern.
4. Repeat the alignment and expose the next damaged marker pattern. Keep repeating until all the damaged marker patterns have been exposed.
5. Remove the sample from the ebeam tool, develop, metalise and lift-off the damaged marker patterns in 10/60 nm Ti/Au metal.
6. Put the substrate back into the ebeam tool and align to the good Penrose markers again but instead of exposing another pattern, collect an image of the damaged marker pattern.
7. Repeat the alignment and collect an image of the next damaged marker pattern. Keep repeating until all the damaged marker patterns have been imaged.
8. Correlate each of the collected images against the same reference image.

Multiple alignments are performed during the exposure and the imaging of the substrate to reduce the effects of drift. Since the damaged marker patterns each take less than 8 s to expose, and the worst case drift rate of our VB6 is approximately 10 nm/min, the positional error of the pattern caused by drift is less then 0.134 nm. Similarly it takes approximately 10 s to collect a 100x100 pixel image so the absolute maximum error introduced by any drift in the writing or imaging is expected to be 0.3 nm.

5.2 Results

The performance of the different types of damaged markers has been investigated as described above. For the additive defects the imaging was performed in a Vistec EBPG5000+

ES HR running at 50 kV in collaboration with Doctor Kevin Lister of the Center of Micro-NanoTechnology, École Polytechnique Fédéral du Lausanne and Hans Romijn of Vistec Lithography BV [121]. The results are shown in figure 5.8. Figure 5.8 a) shows the effect that increasing the density of defects has on the accuracy with which the marker searches can be performed. The plotted values are the average of the standard deviations across all sizes of defects as the density of the defects varies. It is clear to see that as the density increase so too does the standard deviation of the marker searches and therefore the accuracy of any alignment would decrease. It is interesting to note that the errors only increase by ∼1.5 nm as the density changes across the full range illustrated in figure 5.8 a), and even in the worst-case is still less than 3 nm.

In figure 5.8 b) the standard deviation of marker searches is plotted as the size of the additional defects is increased from 10 nm to 600 nm. Again the plotted values are an average over all densities of defect as the defect size increases. It is not too surprising that the accuracy should reduce as the defect size increases, however it is interesting that the measured standard deviation remains consistent at about 1.5 nm until the defects reach 100 nm in size and then there is a large change in the measured accuracy. Again the increase is only about 2.5 nm for the worst-case of 600 nm defects and the marker search accuracy is still better than 5 nm.

To calculate the standard deviations shown in figure 5.8, a 5x5 array of each type of damaged marker was imaged and measured. The markers within each 5x5 array had a randomly generated defect pattern with the specified defect size and density properties, so no two markers were identical. This allowed the standard deviation for each defect size and density to be calculated, since the measurements were statistically independent.

The marker patterns with missing elements were imaged in a Vistec VB6 UHR EWF and we see a similar trend with more severely damaged markers producing larger errors and less accurate marker searches. Figure 5.9 shows the standard deviation of the marker search results for this type of markers. The marker numbers on the x-axis correspond

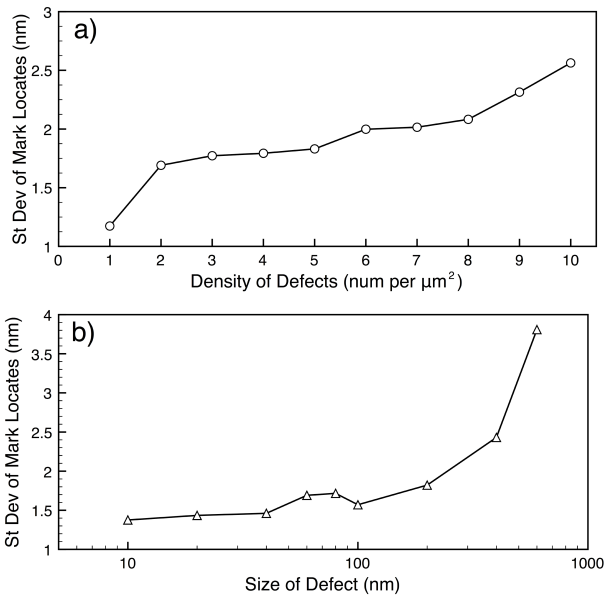


Figure 5.8: Standard deviation of marker searches as, a) the number of, and, b) the size of, the additional defects within the marker area is varied. The standard deviations are calculated from 25 independent measurements per point.

with those shown previously on the marker images in figure 5.6. It is interesting to see that the marker search standard deviation remains roughly constant while there is still something of the marker present. However once the image becomes blank the marker search results essentially become random and there would be no alignment possible.

There is an interesting question of how uniquely defined the marker position is, as the imaged region of the marker patterns become more damaged with less of original pattern remaining intact. Section 3.1.7 discussed the minimum interrogation area need to uniquely identify the position within a larger pattern, and showed that interrogating at least 6% of the total pattern area was enough to uniquely determine the interrogation area's position. A similar result is expected when the marker is subjected to damage. The interrogation area could always be increased to ensure uniquity of position even with severely damaged marker patterns.

To prove that these marker search results equate to real alignment performance we

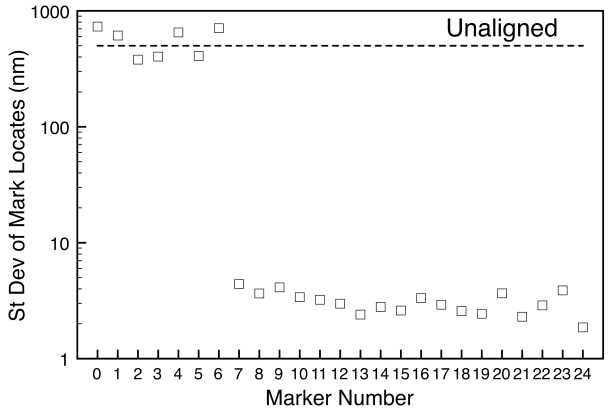


Figure 5.9: Standard deviation of marker searches for the incomplete markers. Marker completeness is proportional to marker number with 0 being nothing to 24 being perfect.

have used sets of the underexposed markers to align and fabricate 1 nm scale Vernier structures in two halves, as a measure of the alignment possible. Four examples are shown in figure 5.10, the first a) with a perfect marker, b) with a good Penrose marker, the third c) with the most damaged marker that still produced a sensible alignment, and lastly d) with one of the markers that gave unaligned results in the marker searches. The alignment errors, as measured by the 1 nm resolution Verniers, was as given in table 5.2. These results confirm that structures with sub-5 nm positional errors can be fabricated by aligning to these heavily damaged Penrose markers.

Marker number	Case in fig. 5.10	x alignment error	y alignment error
22	a)	0±0.5 nm	0±0.5 nm
12	b)	0±0.5 nm	0±0.5 nm
7	c)	0±0.5 nm	0±0.5 nm
0	d)	110±1 nm	33±1 nm

Table 5.2: Alignment errors, as measured from the 1 nm resolution Vernier structures written by alignment using four underexposed markers of various degree of completeness.

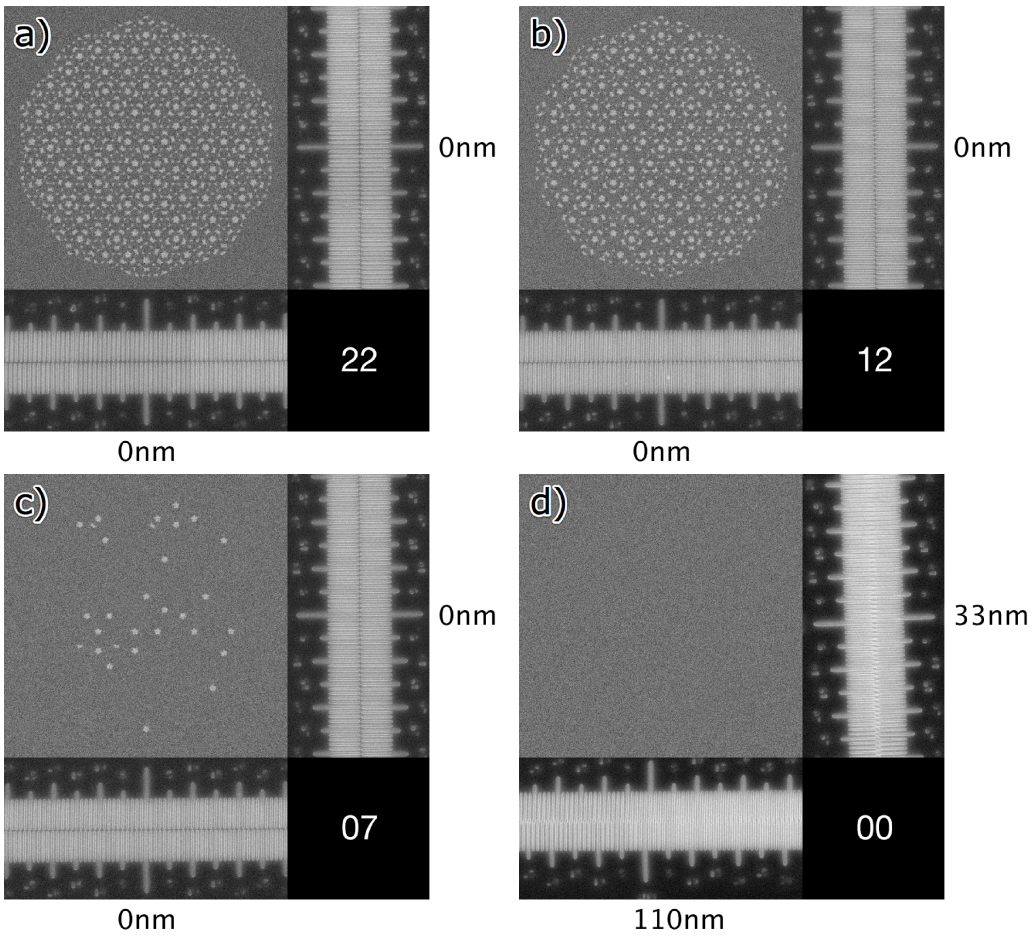


Figure 5.10: SEM images of the fabricated aligned Verniers structures beside a typical marker of that dose. The marker number is written in the bottom-right corner and the alignment indicated by the Verniers is written beside them.

5.3 Conclusions

We have shown the high robustness of correlation-based alignment with Penrose patterns to two significant types of marker damage. With additional defects replacing significant parts of the pattern we have shown that marker searches with a standard deviation of well below 5 nm can repeatability be performed. When sections of the Penrose marker pattern are removed through underexposure we again demonstrate the high robustness of the correlation algorithm, showing sub-5 nm marker searches. Finally, fabricated aligned Vernier structures with alignment errors of less than 5 nm have shown the alignment performance of damaged Penrose markers when used in the writing of physical patterns.

6 Fabricated Structures

The improvements in alignment demonstrated have obvious uses in nanofabrication. In particular, improved alignment allows the fabrication of structures that previously would have been very challenging or even impossible due to physical limitations. A problem with resist is that, particularly for subtractive processes, it has two jobs to perform and in some cases this can lead to compromises in the performance at either task. The first is as a photo/electron sensitive layer and ideally would have infinite contrast and resolution and not suffer from mechanical collapse, suggesting that a thin layer is advantageous. The second is during the pattern transfer step where the resist must be sturdy enough to survive any etch processes, suggesting that a thicker layer is needed.

For lift-off type processes the smallest structures that can be created are limited by the mechanical strength of the resist before it collapses. With the improved alignment process the resolution of the resist can be de-coupled from the smallest features that can be created [122]. This works by separating the desired structure into a series of edges each created in their own layer of resist. Forming the subsequent edges of the structure in other layers of resist then relies just on the accuracy of the alignment of the edges to give the final feature size. The 'resolution' or minimum feature which can be created this way, is now limited by the line edge roughness, the reproducibility and the alignment accuracy rather than the mechanical properties of the resist.

The simplest structures formed using this technique are made from two edges, and are gaps or wires depending on whether the pattern transfer steps are additive or subtractive

respectively. Structures of these types have been fabricated and analysed and in the next two sections they will be examined in more detail. Using three or more edges it would be possible to create more complex structures however this becomes increasingly challenging since each additional steps not only adds errors due to the alignment, but also introduces an additional pattern transfer step and the associated errors. So while it is envisioned that posts and holes, as well as 3, 4 or more terminal nanogap devices could be fabricated using this technique this has not been performed.

There is also scope for creating smaller period gratings by the aligned exposure of several larger period gratings and this has already been used to fabricate Fresnel zone plates for use as diffractive optical structures [123].

6.1 Nanogaps

Perhaps the simplest structure that can be created with an additive process and multiple aligned exposures is the nanogap device. That is a small gap in a conductive wire, with a gap width of the order of a few nanometers. Nanogaps are of great interest in the field of molecular electronics, where they can be used to study the electrical properties of molecules. They can also be used to gain information about the formation of chemical bonds between molecules and the metallic conductors and for studying the conduction properties of individual molecules [124]. Molecular conduction falls off sharply as the length of the molecule under test increases and it is therefore important to reduce the gap size to below 5 nm. The direct electron beam lithographic fabrication of devices at this scale has proven challenging, partly because the conventionally used resists are subject to mechanical collapse and instability at this scale, but also due to the limited accuracy of conventional alignment methods which has ruled out the use of multi-step aligned lithographic processes.

To overcome these limitations several novel fabrication schemes have been proposed, including the use of mechanically controllable breakjunctions [125], atomic force mi-

croscopy [126] and by using accurate control of deposited layers [9]. Although sub-5 nm gaps have been created using these methods, they all put severe constraints on the materials which can be used and on the versatility of the fabricated device, e.g. in terms of how it can be handled. There have also been attempts to work around the limitations of alignment accuracy by using statistical alignment methods [127], however this relies on time-consuming SEM inspection, and is only practical for the production of small numbers of functional devices.

The improved alignment accuracy that comes from using a correlation based alignment procedure with Penrose patterns allows nanogaps with separations of less than 5 nm to be reliably fabricated in a controlled manner. Furthermore, the process described below allows many nanogaps to be fabricated concurrently and so would allow wafer-scale fabrication of sub-5 nm nanogaps.

6.1.1 Process Flow

Nanogap devices have been created using a process based on 4 lithographic levels. The process flow is illustrated in figure 6.1. The starting material is a blank 3" n^- Si wafer, (100) oriented. This is initially thermally oxidised so that an insulating planar substrate is formed onto which the devices will be fabricated. The silicon plays no active part in the operation of the device. The first lithographic layer is used to form the alignment markers that will be used throughout the rest of the processing. The marker pattern contains four distinct types of features, firstly, large crosses that are visible optically act as reference points when the wafers are mounted in the ebeam tool. Secondly, 20 μm diameter octagons are created which will be used for global alignment using the built in alignment routines of the ebeam tool. These are positioned in the centre and at four sites towards the North-East, South-East, South-West and North-West edges of the wafer. Thirdly, in the four corners surrounding each 5 mm by 5 mm cell there are sets of 10 μm diameter octagons which will be used for cell alignment. Fourthly, sets of Penrose pattern markers are also positioned in the four corners of each 5 mm by 5 mm cell, for the

precise device alignment. The process typically used for the markers level is as follows:

Clean substrate	- 5 min acetone, 5 min IPA, with ultrasonic agitation.
Spin Resist	- 4% 2010 PMMA 5 krpm 60 s. 137 °C hot plate bake, 180 s.
	- 2.5% 2041 PMMA 5 krpm 60 s. 137 °C hot plate bake, 180 s.
Exposure	- Global markers: Dose 700 μCcm^{-2}, Resolution 1.25 nm, BSS 20 nm, Beam 64 nA
	- Penrose markers: Dose 800 μCcm^{-2}, Resolution 1.25 nm, BSS 5 nm, Beam 1 nA.
Develop	- 1:2.5 MiBK:IPA, 23 °C, 30 s. IPA rinse.
Ash	- 40 W, 30 s.
Metallise	- Ti 10 nm, Au 60 nm.
Lift-off	- 40 °C acetone, 2 hr, IPA rinse.

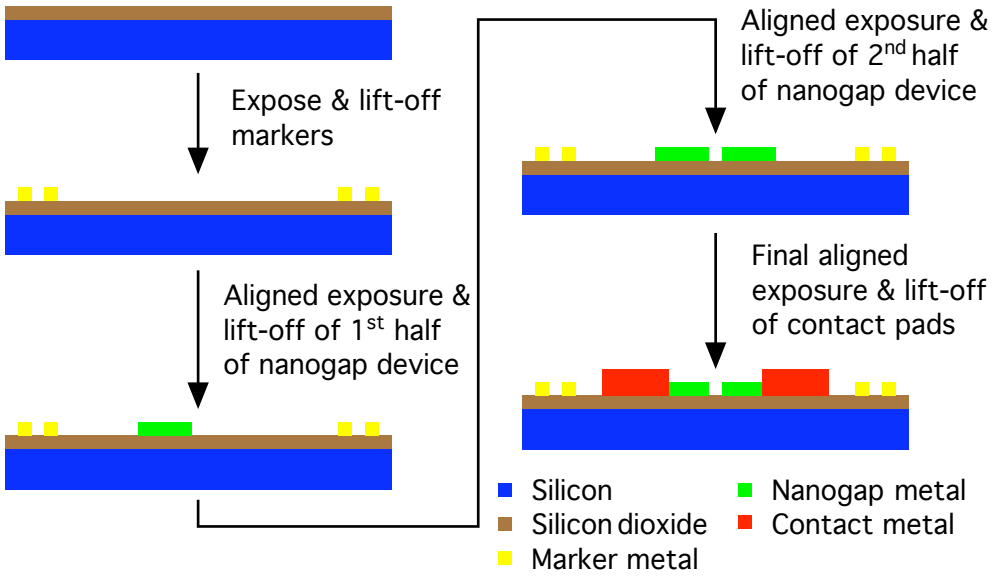


Figure 6.1: Schematic illustration of the main fabrication steps in the process used to make aligned nanogap structures. *(not to scale)*

Next the first half of the nanogap device is written aligned using these markers, the large octagons are used as global alignment markers to roughly align the wafer, the global

alignment has an accuracy of about 20 nm. Then the Penrose pattern cell markers are used for the precise alignment of the nanogap structure with respect to those markers. Again a standard PMMA bi-layer is used however, since thinner metal will be used, thinner resist is also used to reduce patterning distortions caused by forward scattering. The process typically used to fabricate the first part of the nanogap structure is as follows:

Spin Resist	- 2.5% 2010 PMMA 5 krpm 60 s. 137 °C hot plate bake, 180 s.
	- 2.5% 2041 PMMA 5 krpm 60 s. 137 °C hot plate bake, 180 s.
Exposure	- Dose 720 μCcm^{-2}, Resolution 0.5 nm, BSS 4 nm, Beam 1 nA
Develop	- 1:2.5 MiBK:IPA, 23 °C, 30 s. IPA rinse.
Ash	- 40 W, 30 s.
Metallise	- Ti 2 nm, Au 10 nm.
Lift-off	- 40 °C acetone, 2 hr, IPA rinse.

The second half of the device is written, again aligned to markers from the first exposure, however since the markers used during the definition of the first half of the nanogap device will now be partly covered in metal, a second set of markers was used to avoid the loss of accuracy associated with incomplete metal covering the marker as discussed in section 5.1.2. Although the use of different markers is not ideal, the two sets of Penrose markers used are written sequentially during the initial exposure to minimise the effect of drift and the exposure time, at less than 2 s is such that even in the worst case of a drift rate of 10 nm/min the error induced would be much less than 0.3 nm. An identical process is used to that for the fabrication of the first half of the nanogap structure.

Finally the contact pads were written in a thicker PMMA bi-layer to allow a thicker metal layer to be used. The contact pads were aligned using the 10 μm octagonal cell markers since there was a much greater tolerance on the alignment. They were fabricated by lifting-off a thicker layer of evaporated metal (50nm Ti / 150nm Au) to permit bonding or probing. The process flow for the contact pads is as follows:

Spin Resist	- 8% 2010 PMMA 5 krpm 60 s. 180 °C oven bake, 1 hr.
	- 2.5% 2041 PMMA 5 krpm 60 s. 180 °C oven bake, 2 hr.
Exposure	- Dose 600 μCcm^{-2}, Resolution 1.25 nm, BSS 40 nm, Beam 32 nA
Develop	- 1:2.5 MiBK:IPA, 23 °C, 30 s. IPA rinse.
Ash	- 40 W, 30 s.
Metallise	- Ti 50 nm, Au 150 nm.
Lift-off	- 40 °C acetone, 2 hr, IPA rinse.

The versatility of this process is a major advantage, as it is simple to fabricate the two halves of the nanogap from different metals which, for instance, could allow for molecules with different functional ends to be correctly orientated when bonding to the gap. It is also relatively simple to extend the process to add electrodes close to the sides of the nanogap to act as gate electrodes. Furthermore, the silicon substrate or a metal underlayer, deposited before the oxidisation, could also be used as a gate.

The process outlined above has been used to create arrays of nanogap devices rather than single nanogaps. Slightly varying the designed gap size between columns of these arrays allows the remaining bias in the fabrication process associated with variations in the exposure dose and deposition to be accounted for. With one SEM inspection step the column with the best range of gaps can be selected and all gaps in equivalent columns connected to the pads for electrical investigation.

Within each column of the array there are 10 nanogaps designed to have increasing gap widths in 0.5 nm steps. This allows a measure of the repeatability and accuracy of the fabrication process to be obtained. Measuring the gap width along one column should show a gradually reducing trend with a slope related to the 0.5 nm steps between gaps, from this a measure of the variations due to the fabrication process can be obtained. Measuring this same data at several sites on a single wafer and on multiple wafers gives a measure of the variation due to alignment inaccuracies and the overall repeatability of the process.

6.1.2 Results

Several wafers with multiple nanogap device arrays have been fabricated using a Vistec VB6 UHR EWF ebeam tool. Figure 6.2 shows SEM pictures of one column of nanogap devices after the whole fabrication process has been completed. The gaps shown in figure 6.2 have widths ranging from 9 nm down to 3.5 nm. It was estimated that distances on the micrographs could be determined to within 10 pixels, corresponding to 0.75 nm, so the measurement error was estimated to be 0.75 nm.

These measurements have been plotted on the graph of figure 6.3 and a least squares fitted straight line has been overlaid on the data to get a measure of the actual gap size decrement. This straight-line model has a gradient of -0.42 nm which is in good agreement with the designed step size of 0.5 nm.

Similar measurements from another site on the same wafer and from another wafer are plotted in figures 6.4 and 6.5 and similar straight-line models have been fitted to that data. The gradients from all three measurements are in good agreement with each other and with the design value of -0.5 nm. Further analysis shows that the standard deviation of the gap size across all three samples is less than 1.2 nm.

6.1.3 Molecular Conduction Results

Professor Richard Whitby and his group at the University of Southampton have collaborated with this work and have performed some preliminary measurements on the conduction of the nanogap devices. They measured the conductivity of each of the ten gaps on a single chip using a Keithely 2636 dual-channel source meter which in this setting has a current sensing resolution of approximately 100 fA. The set-up is shown schematically in figure 6.6.

Using this set-up allows in-situ measurements to be made of the nanogap devices under solution, with the aim being to be able to measure dynamic molecular conduction events as multiple molecular wires bridge the gap between the nanogap electrodes.

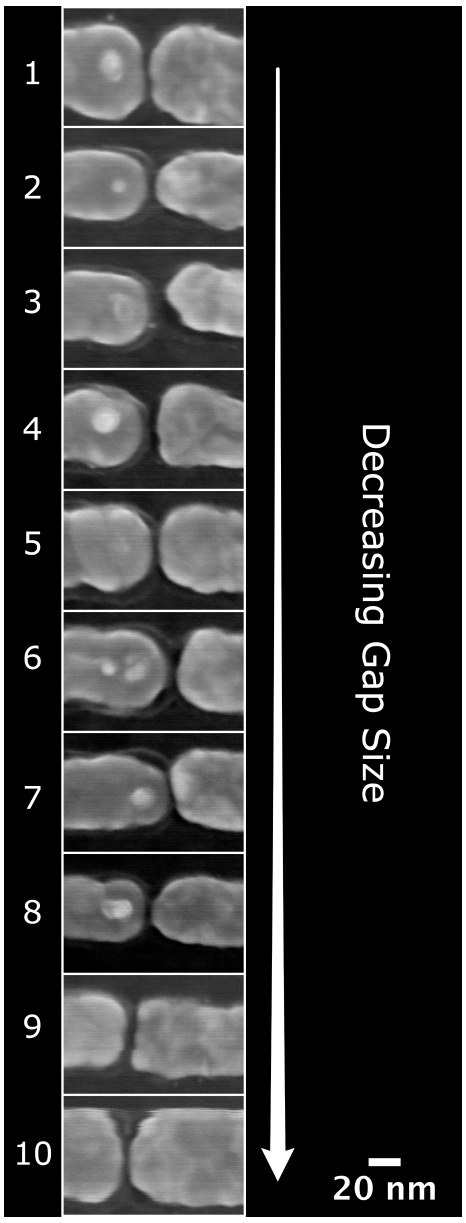


Figure 6.2: Typical example of the fabricated nanogaps, showing one column of the array with gradually reducing designed gap width from top to bottom in 0.5 nm steps.

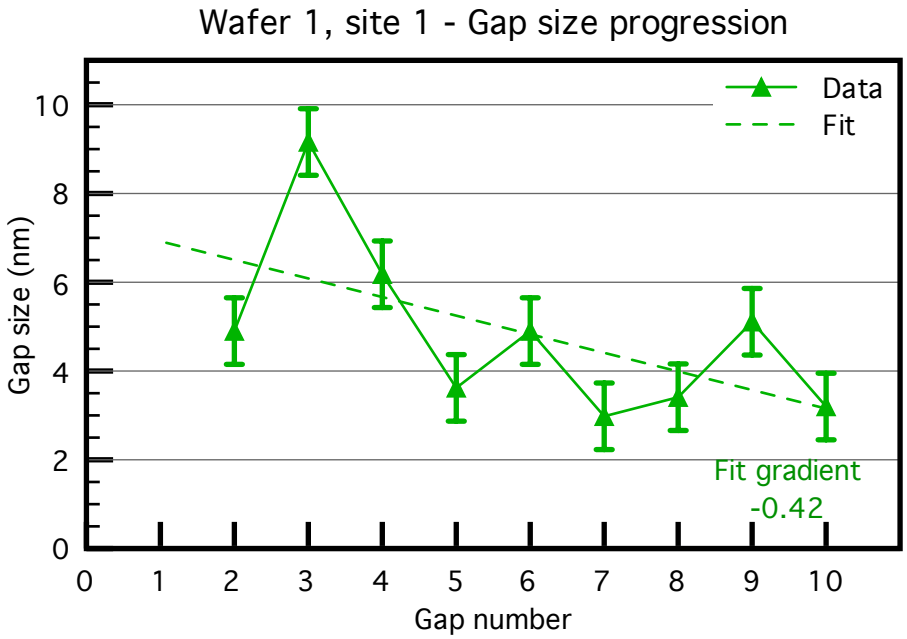


Figure 6.3: Measurement of the gap widths from the SEM images shown in figure 6.2 with a least-squares fitted straight-line.

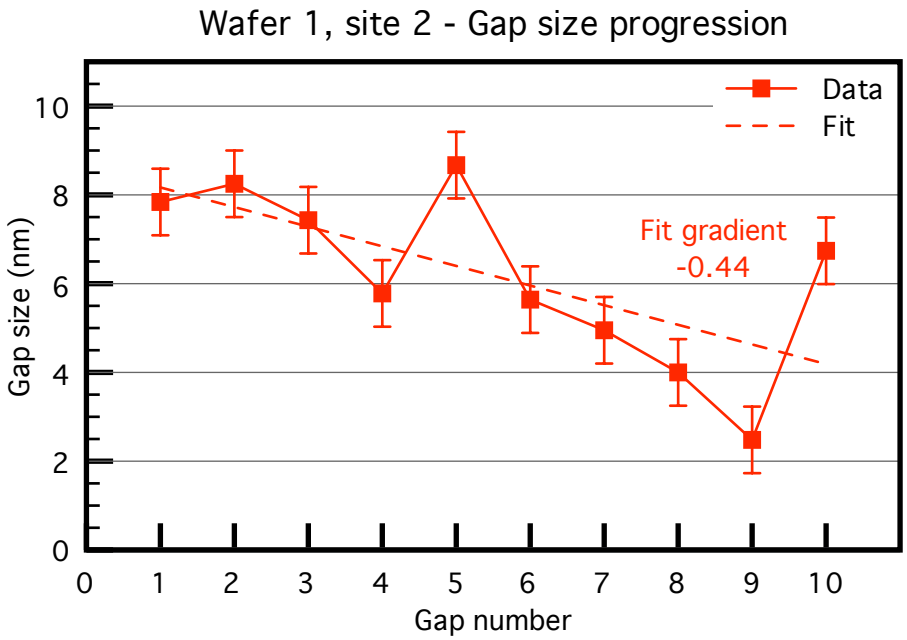


Figure 6.4: Measurement of the gap widths from the second site on wafer 1 with a least-squares fitted straight-line.

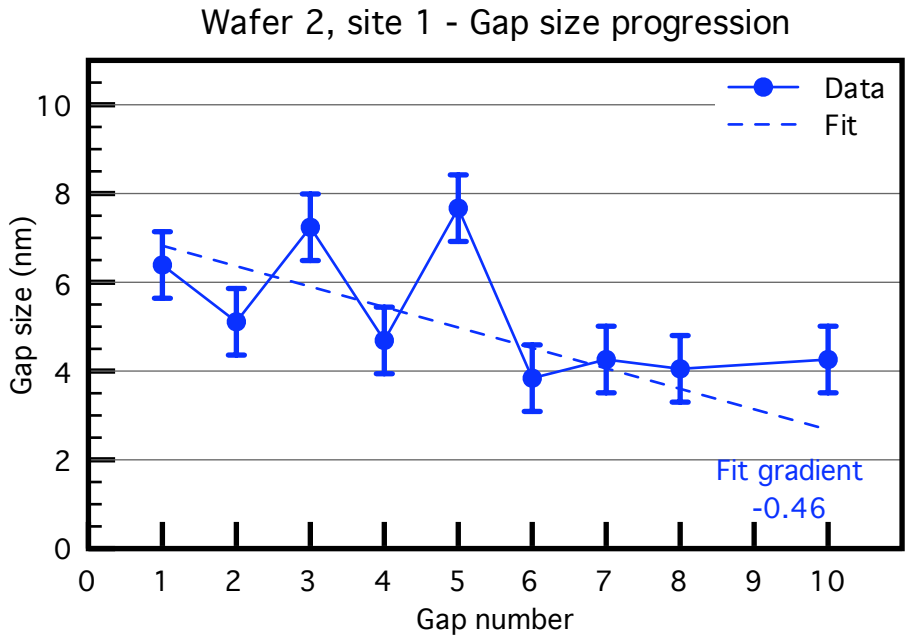


Figure 6.5: Measurement of the gap widths from the first site on wafer 2 with a least-squares fitted straight-line.

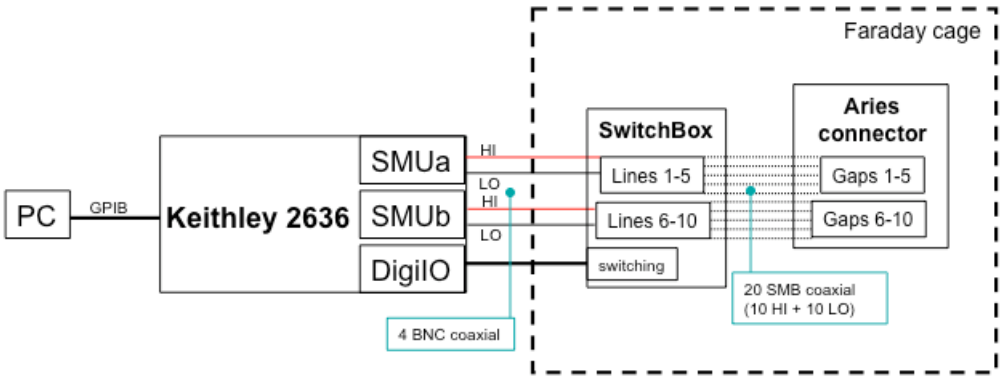


Figure 6.6: Schematic of the equipment set-up used to measure the conduction of the nanogap samples.

Initially the conduction of the ten nanogap was measured dry. The resulting conductivity is shown in figure 6.7. The conductivity of the gaps is relatively close to the background noise level and shows that this measured leakage current is small enough to allow molecular wire assembly events to be measured as they are expected to be of the order of a few hundred pA. The leakage current is also close to the 100 fA minimum current that could reliably be measured with this set-up.

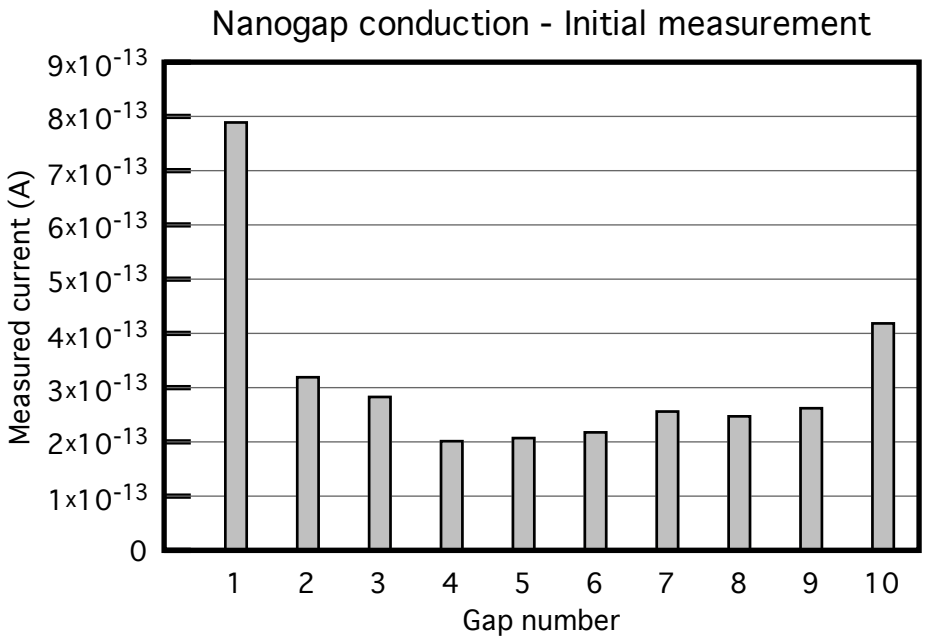


Figure 6.7: Conductivity of the nanogaps as initially measured on the bare nanogap sample.

Secondly a solvent cell was added over the nanogap sample and the conductivity of the nanogaps was measured while submersed in a 1:1 soultion of the solvents Acetonitrile (Methyl cyanide or MeCN) and Dichloromethane (DCM). As shown in the measurements in figure 6.8, the presence of the solvent increases the conductivity of the gaps, but more significantly the profile of conductivity across the gaps matches the designed gap width variation from smallest to largest gap. (Note that gap number one was conductive, however the current was too large, >10 nA, for the measurement range used.)

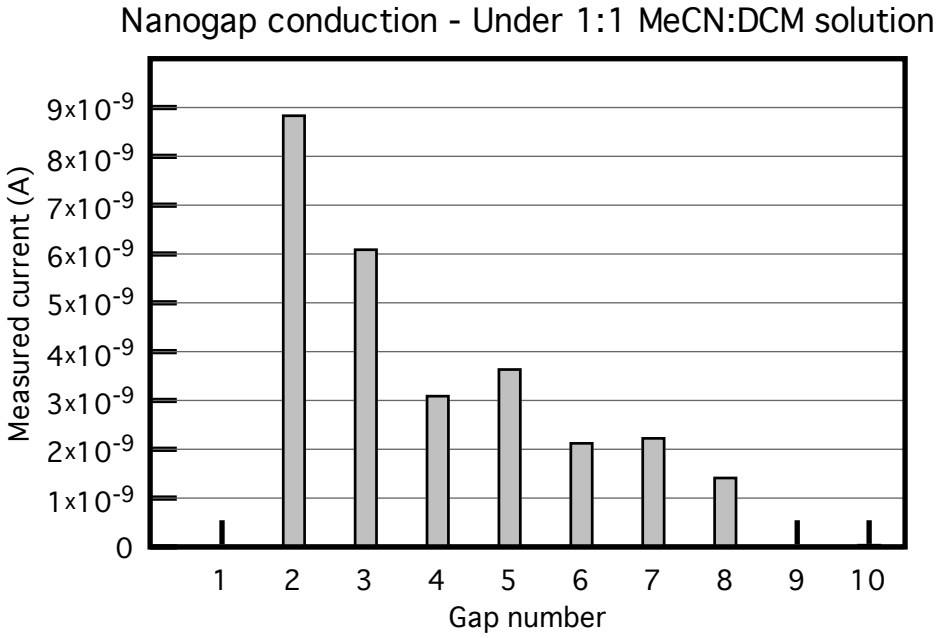


Figure 6.8: Conductivity of the nanogaps measured in situ while the chip is under a 1:1 MeCN:DCM solvent solution. The conductivity profile shows an increase in current over the bare die and matches well with the designed gap size variation. Note that the conductivity of gap one was greater than the maximum of the measurement range used (>10 nA).

The nanogap device sample was then soaked in a 2.5 μM solution of molecular wires "OPEC-5ring-Sac" in DCM for 2 hours and then dried before the conductivity of the nanogaps was measured again, as shown in figure 6.9. An increase in conduction of a few hundred pA was expected but not observed. The small currents observed is most likely due to the vast surface area of gold exposed to the molecular wires and available for them to bind to instead of within the nanogap itself.

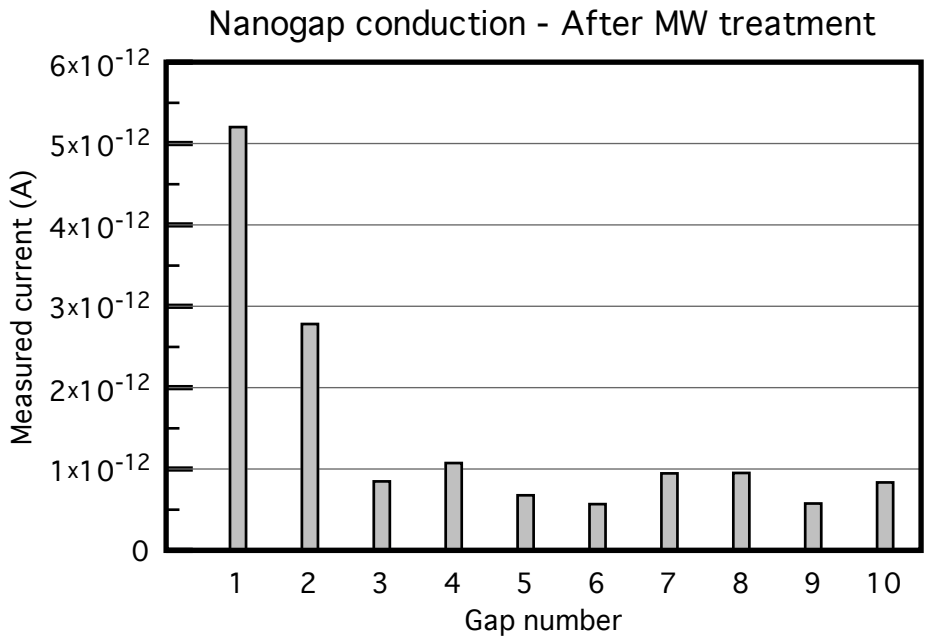


Figure 6.9: Conductivity of the nanogaps after having been treated with the molecular wire solution showing little change with respect to the initial conductivity profile.

6.1.4 Conclusions

A new fabrication scheme for nanogaps has been introduced exploiting the advances in alignment accuracy in electron beam lithography. It has been shown that correlation-based alignment with Penrose pattern markers provides sufficient accuracy to be a feasible method of fabricating nanogap devices. A four-level fabrication process has been used to fabricate several arrays of nanogap devices and it has been demonstrated to be a reliable,

repeatable method to produce sub-5 nm nanogaps with a random variation in gap size of just 1.2 nm measured at three sites across two wafers.

The measurement of the conductivity of a set of nanogap devices has confirmed that they provide a low-leakage structure which could be used to measure molecular wire assembly events. The measurements of conductivity in the presence of a solvent has shown a good correlation between measured conductivity and the designed gap width. The lack of conclusive evidence for conduction through molecular wires is likely due to a lack of molecular wires binding in the nanogaps themselves caused by the molecular wires binding to the large surface area of gold exposed to the molecular wire solution.

The method has been shown to allow small gaps to be fabricated, in quantity, on a firm substrate which can then be shipped without destroying the fabricated structures for further electrical measurements or study, something which cannot be done with nanogaps formed using mechanically controllable break junctions for example.

6.2 Nanowires

The simplest structures that could be created using this method and an additive pattern transfer method are nanogaps, as discussed in the previous section. The equivalent when considering a subtractive pattern transfer process is the creation of nanowires, that is thin metallic structures with widths on the order of a few nm. The fabrication of extremely thin metallic wires, or nanowires, has applications in reducing the gate length of advanced transistors, allowing an increase in operating frequency. Perhaps more interestingly but also more challenging, the ability to fabricate wires that have atomic widths opens up the possibility of studying the effects of quantised conduction in one-dimensional electron flows along atom chains. Traditionally this has been investigated through the use of mechanically controllable break junctions or by scanning probe measurements. These techniques limit the range of materials that can be used and rely on a dynamic process to establish the conducting channel making the structures inherently unstable (i.e. when

the force is removed from the break junction, or the SPM moved, the conduction channel is lost).

Similar to the nanogap devices, the limiting factor in conventional processing is the finite resolution of the resist. Furthermore the finite grain size of the evaporated metal also limits the minimum width of wire that can be created by a lift-off process. Forming the wires using a subtractive process allows the metal to be deposited by sputtering or epitaxy rather than evaporation which provides finer control over the properties of the deposited film. Varying deposition parameters such as the gas pressure, sputtering power and substrate temperature allows the grain size, texture or step coverage of the film to be finely tuned to give a metal layer with the desired properties.

6.2.1 Process Flow

Nanowire devices have been created using a very similar process to that used for the nanogap devices described above. Again the process is based on four lithographic exposures as illustrated in figure 6.10. Starting with a blank 3" n^- Si wafer (100) orientation, a layer of either silicon dioxide or nitride is grown or deposited to give an insulating planar substrate on which the devices may be formed. The first exposure is to create the alignment markers. The exposure and process are identical to that described above for the nanogap devices.

In the next step a thin layer of metal is deposited across the entire wafer, this will be formed into the nanowire structures by two subsequent etch processes. Tungsten was chosen as previous work [128] had developed a slow dry-etch with vertical side walls, and also because the Plassys MP900S sputter tool had previously been set-up with recipes to deposit thin films of tungsten with low stress. A 6 nm thick layer of tungsten was deposited using the low stress process conditions. Although no detailed measurements were taken, the inspection by SEM of subsequent steps in the process revealed that either the grain size of the deposited tungsten was below the resolution limit of the SEM and therefore caused no issues, or that the grains produced no significant topography for the

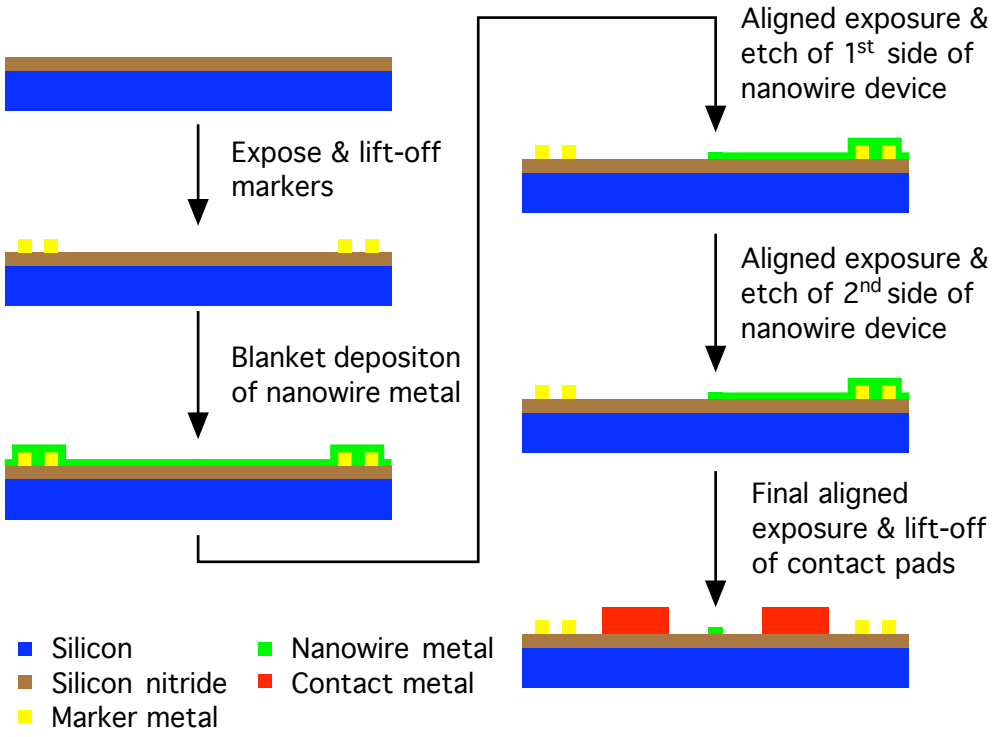


Figure 6.10: Schematic illustration of the main fabrication steps in the process used to make aligned nanowire structures. Note that a connection is made between the nanowire and the contact pads out of the plane of the page. *(not to scale)*

SEM to detect.

Once the tungsten had been deposited the first layer of resist was spun on the wafer and the first half of the nanogap device was written aligned using the markers; the large octagons as global alignment markers and then the Penrose pattern cell markers for the precise alignment of the nanowire structures with respect to those markers. Whereas a positive resist was used for the nanogap devices since only a small opening was required here we wish to protect a small area from the subsequent etch so a negative resist was used. Two different resists were used during the development of this process, firstly, a conventional negative resist, hydrogen silsesquioxane (HSQ) was used. The process flow for the fabrication of the first part of the nanowire structures with HSQ was as follows:

Clean substrate - 5 min acetone, 5 min IPA, with ultrasonic agitation.

Ash	- 200 W, 5 min.
Spin Resist	- 3:1 MiBK:HSQ 4 krpm 60 s. 80 °C hot plate bake, 120 s.
Exposure	- Dose 1000 μCcm^{-2}, Resolution 0.5 nm, BSS 2 nm, Beam 1 nA
Develop	- 25% TMAH in RO water, 23 °C, 30 s. RO water then IPA rinse.
Etch	- SF_6/N_2 5/55 sccm, 120 s.
Resist strip	- 10:1 Silicon dioxide etch, 60 s.

In subsequent experiments, PMMA resist, turned negative through the use of a very high exposure dose, was used. The process flow for the fabrication of the nanowire structures using negative PMMA was as follows:

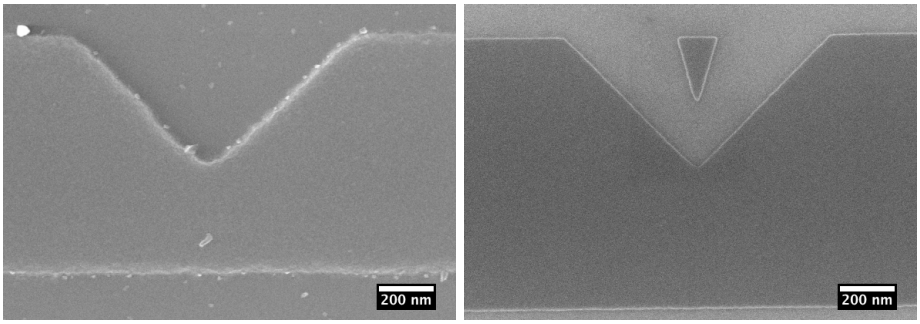
Clean substrate	- 5 min acetone, 5 min IPA, with ultrasonic agitation.
Spin Resist	- 2.5% 2041 PMMA 5 krpm 60 s. 137 °C hot plate bake, 120 s.
Exposure	- Dose 120, 000 μCcm^{-2}, Resolution 0.5 nm, BSS 2 nm, Beam 2 nA
Develop	- 1:2.5 MiBK:IPA, 23 °C, 30 s. IPA rinse.
Etch	- SF_6/N_2 5/55 sccm, 120 s.
Resist strip	- Ash 200 W, 5 min.

The resist, once developed, is used as the mask so the pattern could be transferred to the substrate by dry etching using SF_6. Finally, the resist layer is completely removed from the substrate and fresh resist is applied in preparation for the exposure of the second half of the device. This is written in a similar fashion to the first half of the structure, the only difference being that a second set of markers is used during the alignment as in the case of the nanogap fabrication.

Using HSQ resist to define the nanowire device required the use of hydrofluoric acid (HF) to remove the resist since exposed HSQ is very similar to SiO_2 in properties. In this case the substrate was prepared with Si_3N_4 as this has a much lower etch rate in HF than SiO_2. Although the etch rate of tungsten in HF, as measured by Williams, et

al. [129], is <2 nm/min this was enough to attack the thin layer of tungsten, such that the tungsten film was rendered non-conductive.

To overcome this limitation a switch was made to using PMMA resist. As previously noted, in section 2.1.2, when PMMA is exposed using a large electron dose it has the properties of a negative resist. While PMMA exposed at a large enough dose to turn negative is more stubborn to remove, it can still be removed in an O_2 plasma such as in a barrel asher at moderate power. This O_2 plasma had no measured effect on the W film and there was the added benefit of improved pattern fidelity over the HSQ patterning when using negative PMMA, as can be seen in the comparison of figure 6.11. Since there was no longer an HF etch to remove the resist a switch was made to wafers with SiO_2 coatings rather than Si_3N_4 to allow the same type of wafers to be used in the fabrication of both nanogap and nanowire devices.



(a) Using HSQ resist. (b) Using negative PMMA resist.

Figure 6.11: First level of nanowire structure after the W has been etched but before the resist has been removed. The resist used in 6.11(a) is HSQ and in 6.11(b) is negative PMMA. The improvements in the pattern definition through using negative PMMA can be seen.

The fabrication is completed by writing the contact pads in a thicker positively exposed PMMA bi-layer, aligned using the octagonal cell markers, and then transferred, by lift-off, into a thicker layer of evaporated metal (50nm Ti / 150nm Au) to permit bonding or probing. The process flow is identical to that used for the fabrication of the contact pads in the nanogap process.

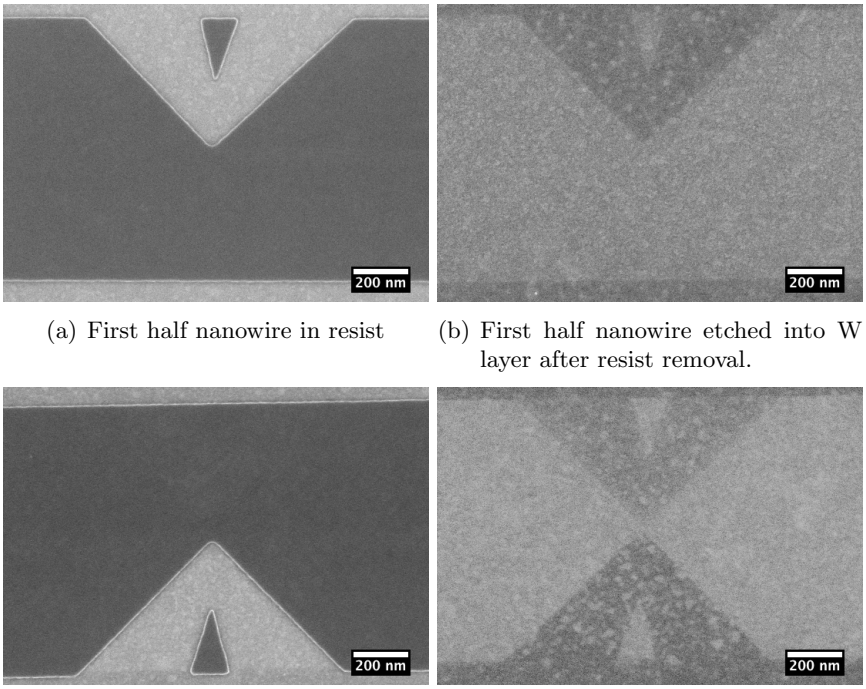
This process has been used to create arrays of nanowire devices, again with slightly varying designed wire sizes between the columns of these arrays to allow the remaining bias in the fabrication process associated with the exposure dose and etching to be accounted for. With one SEM inspection step the column with the best range of wire widths was selected and connected to the pads for electrical investigation.

Within each column of the array there are ten nanowires designed to have wire widths increasing in 0.5 nm steps. This allows a measure of the repeatability and accuracy of the fabrication process to be obtained. Measuring the wire width along one column should show a gradually reducing trend with a slope related to the 0.5 nm steps between wire widths, from this we can get a measure of the variations due to the fabrication process. Measuring this same data at several sites on a single wafer and on multiple wafers allows a measure of the variations due to the alignment inaccuracies and indeed the overall repeatability of the process to be obtained.

6.2.2 Results

Figure 6.12 shows a series of four SEM images of a nanowire device at four key points during the fabrication process. It shows that the desired patterns can be formed in the negative PMMA resist, and further that this resist can be used in the dry etch process to transfer the pattern to the underlying tungsten layer. The pattern is transferred cleanly with little degradation to the definition of the edges of the structure, however some roughening of the oxide surface is observed after the dry etching. It can also be seen that there is very little undercutting during the etch which would mean that this method of pattern transfer should be compatible with the fabrication of sub-10 nm structures.

The second pair of images show that the second half of the nanowire structure can also be formed in resist and transferred into the tungsten layer with similar performance to that seen in the first half of the nanowire fabrication process. Secondly, the images also demonstrate that the alignment process allows structures in the sub-10 nm regime to be achieved.



(a) First half nanowire in resist

(b) First half nanowire etched into W layer after resist removal.

(c) Second half nanowire in resist

(d) Second half nanowire etched into W layer after resist removal.

Figure 6.12: SEM micrographs of a nanowire device at the four key stages of the fabrication process.

An example of the range of nanowire devices that have been fabricated on a single chip is shown in figure 6.13.

Electrical measurements were performed to measure the conductance of a set of fabricated wires. A semiconductor parameter analyser was used to measure the current passed along each wire as the drive voltage applied across the wire was swept from 0 to 100 mV. From the current and voltage measurements the resistance of each wire was calculated. The resistance of one set of nanowire devices is shown in figure 6.14. The resistances were calculated from currents measured at a drive voltage of 25 mV. Figure 6.14 shows that as the designed wire width is decreased, the resistance increases, and the fitted trend line shows that the variation is linear and matches the linear decrease in designed wire width.

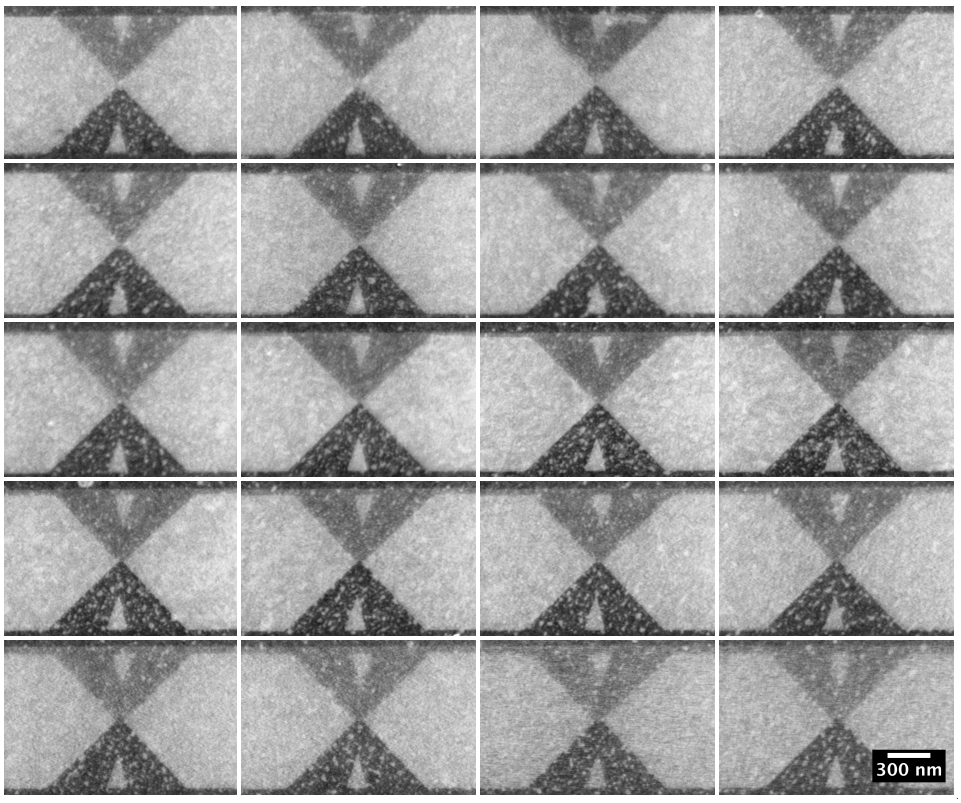


Figure 6.13: Typical example of the fabricated nanowire structures showing a range of wire widths decreasing from left-to-right along each row in steps of 1 nm and from top-to-bottom between rows.

The measured nanowires were imaged in an Hitachi S900 SEM as shown in figure 6.15. The top-left image shows a nanowire with a designed wire width of 27 nm. The widths increase in 1 nm steps moving to the right along each row. The first wire was damaged, probably from electrostatic discharge, and therefore its resistance could not be measured.

6.2.3 Conclusions

The fabrication of sub-10 nm conducting wires using a two-step, subtractive pattering process has been demonstrated. The four step fabrication process has been shown to allow sets of wires of varying widths to be produced and, although it has not been demonstrated, it is expected that a good level of repeatability could be achieved. Electrical measurement

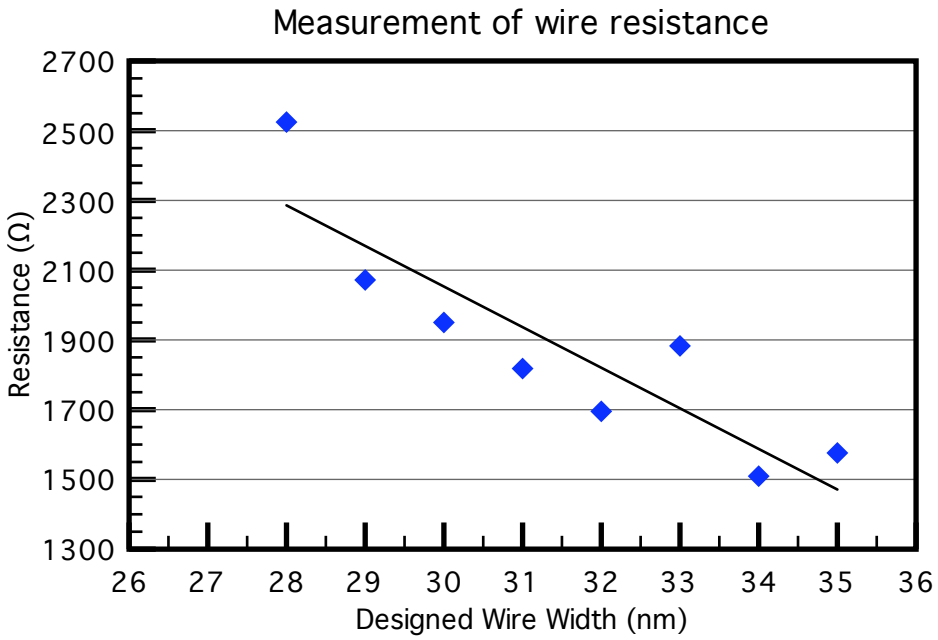


Figure 6.14: Electrical measurements of the resistance of a set of nanowire devices showing the trend in the resistance is consistent with the variation in the designed wire widths.

of the fabricated nanowires has shown that varying the designed wire width has a direct influence on the measured resistance of the fabricated structures. Although further work is required it is believed that the use of correlation-based alignment with Penrose patterns and the fabrication process described could allow the wafer-scale production of sub-10 nm conducting wires.

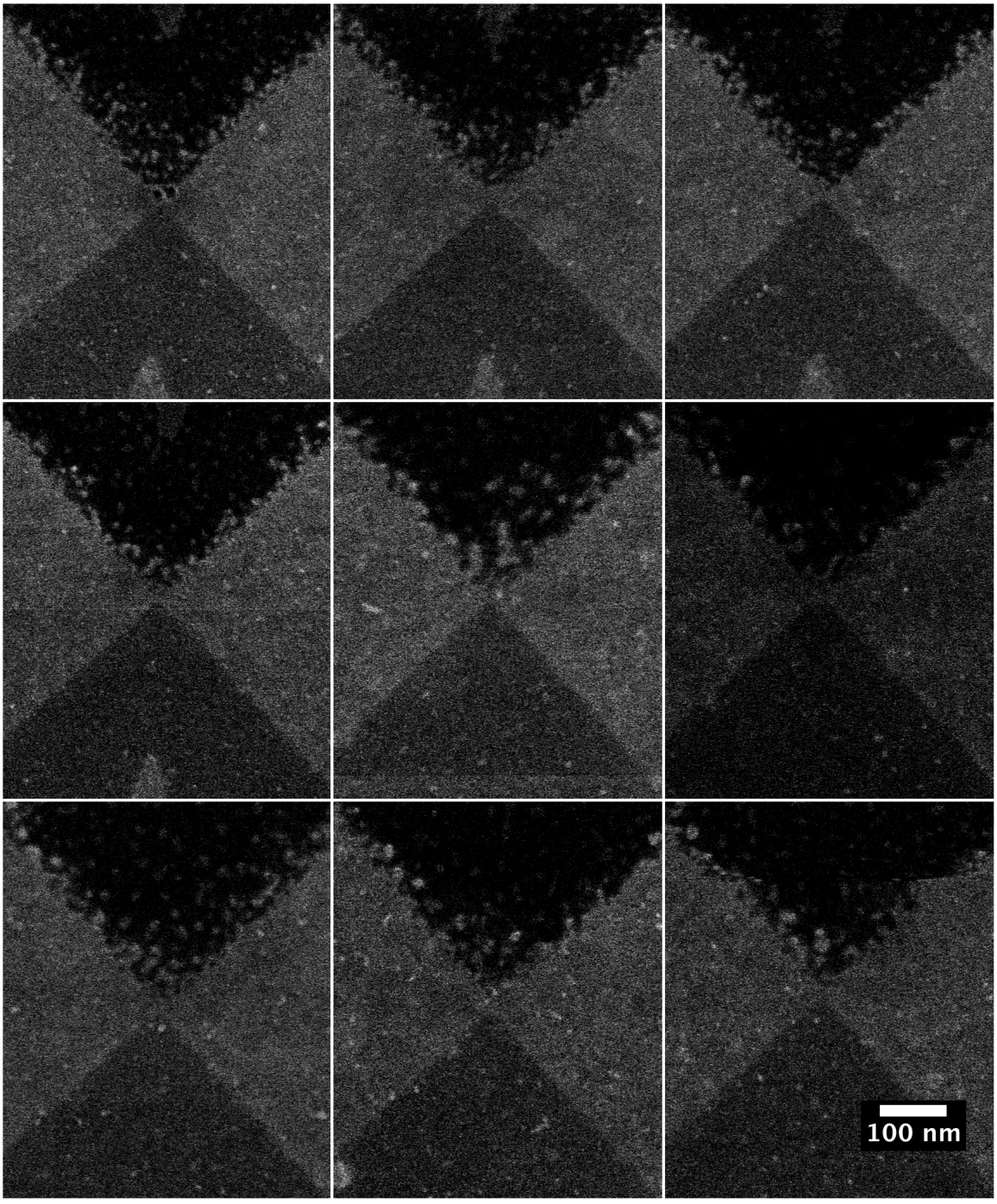


Figure 6.15: SEM images of the measured nanowire devices.

7 Conclusions

7.1 Summary

Alignment is an important process in microfabrication and top-down nanotechnology. Correlation provides a linear, analysable method of detecting markers. Since it acts as a matched filter, correlation is optimum, in terms of maximising the signal-to-noise ratio, for locating a signal in the presence of noise. Correlation based marker location has allowed an analytical study of the alignment process to be performed.

Simple geometrically shaped markers, as used in conventional alignment schemes, encode positional information inefficiently due to the presence of relatively few edges. Using a more complex marker pattern that encodes more positional information allows the alignment accuracy to be improved. A wide range of possible marker patterns have been studied and a range of metrics defined to measure their performance in a correlation based alignment process. PCE has been shown to provide a measure of the peak-sharpness of autocorrelation and the importance of this to alignment has been discussed. From this analysis Penrose pattern were selected as the optimum type of pattern for use as alignment markers in a correlation based alignment process.

Penrose patterns have an inherent aperiodicity which gives them a sharply-peaked autocorrelation function. They exhibit approximately 50% fill of many similarly sized elements, maximising their high spatial frequency components while allowing the size of the marker elements and its extent to be optimised for the fabrication process employed.

Penrose patterns are formed from elements that are incoherent with a regular rectangular sampling array which provides good tolerance to undersampling. Finally, the incoherence with a rectangular sample grid minimises the effect of grid degeneracy on alignment accuracy allowing sub-pixel positional information to be obtained.

A correlation-based alignment process has been implemented on a Vistec VB6 UHR EWF allowing jobs exploiting the improved alignment accuracy to be exposed routinely. The performance of correlation based alignment with Penrose patterns has been experimentally compared to that of conventional alignment. The experimental results showed an order of magnitude improvement in the standard deviation of marker searches provided by correlation based alignment with Penrose patterns. Measurements of the stage drift suggested that the standard deviation of mark locates using the correlation based alignment and Penrose patterns is less than 1.1 nm.

A Fourier transform based method, which can measure alignment accuracy from the aligned fabrication of inter-digitated gratings at the nanometer level, has been introduced. Experimental results show that alignments using Penrose pattern markers and a correlation based alignment process with errors of approximately 0.6 nm can be consistently achieved. This is commensurate with the error in the stage interferometer system, and suggests that this may now be the limiting factor in the alignment process.

A systematic investigation into how defects in the fabrication of Penrose pattern markers affect the attainable alignment accuracy has been performed. This has shown the high robustness of correlation-based alignment with Penrose patterns to two significant types of marker damage. The standard deviation of marker searches was remarkably unaffected by the two type of marker damage studied, remaining sub-5 nm with as much as 80% of the Penrose marker pattern missing. Fabricating aligned Vernier structures with alignment errors of less than 5 nm has demonstrated the alignment accuracy possible, even with severely damaged Penrose markers.

One of the opportunities created by the improved alignment accuracy is the fabri-

cation of nanometer sized gaps and wires using two-stage aligned fabrication processes. These processes avoid the problems of resist collapse and the limited resolution of resist, and instead are limited by the alignment accuracy, line edge roughness and repeatability of the exposure and the pattern transfer process. Nanogaps between Ti/Au contacts with widths as small as 3.5 nm have been fabricated. The experimental results show that ranges of gaps with a standard deviation for the gap width measured at three sites, across two wafers of less than 1.2 nm can be fabricated. One set of nanogap devices have been measured electrically by collaborators at the University of Southampton and it was shown that the conduction through the gap when immersed in a 1:1 MeCN:DCM solvent solution varies directly with the designed gap width. Although a molecular wire solution was applied to the gaps an increase in conduction was not measured suggesting that the molecules bonded to the large areas of exposed gold and did not form conductive paths across the nanogaps.

A two-step, aligned method of fabricating very thin conducting wires has also been demonstrated. The nanowires, formed in a 6 nm thick tungsten film, and covering a range of widths to below 5 nm were successfully fabricated. Electrical measurements on the fabricated nanowires showed that conducting wires could be formed using the two-step etching process and that the resistance of the wires was inversely proportional to the designed wire width.

The demonstrated methods of fabricating nanogaps and nanowires devices, are extremely versatile in terms of the materials that can be used to form the gap and wires, and in the utility of the devices once fabricated. The principle advantage to these methods, however, is the vast number of devices that can be created in a single process, suggesting that two-step aligned fabrication methods could be a viable route for the industrialisation of nanogap and nanowire device fabrication.

7.2 Future Work

The work presented in this thesis could be taken forward in five distinct areas. Firstly, in collaboration with Vistec Lithography BV, and in particular Hans Romijn, work has started to implement the correlation based alignment technique commercially as part of the software system distributed with Vistec's EBPG 5000+ series of electron beam lithography tools. This has involved firstly adding methods to generate Penrose marker patterns and the associated reference images automatically. Secondly implementing a method of using a low-resolution image as a coarse search for rough positioning followed by the correlation alignment with a higher resolution image from the centre of the coarse positioning step for finely determining the marker's position. The routines have presently been incorporated as part of the "BEAMS" control software for the EBPG 5000+ and work is ongoing on improving the speed of the image capturing hardware, and refining the coarse and fine positioning strategy. It would be interesting to see the option of using correlation more widely used in the electron beam lithography community as it adds a versatility to the tools as well as providing the improved accuracy that Penrose patterns offer.

Secondly, to further the study of molecular conduction, the fabrication of the nanogap devices should be extended to include a passivation layer to cover the large gold surfaces away from the nanogap. This would help the molecules to bind to the nanogap region with a vastly increased likelihood of molecules bridging the gap between the contacts. It would be then beneficial to fabricate a range of gap sizes and of molecular wire lengths to investigate if a relationship between the gap size as measured by SEM and the length of molecule which bridge each gap could be established. Further work could then begin to examine the fabrication of three contacts with nanometer sized gaps between them which would allow investigations into allowing the conduction through molecules to be gated by the third electrode or even allow more complex molecular structures to bind to

the contacts for electrical measurement.

Thirdly, nanowire devices, as detailed in chapter 6, have several interesting uses none of which have even begun to be investigated here. In particular, although conductance has been examined, quantisation of current flow was not observed suggesting that refinements to the fabrication process may be necessary to produce thin enough wires to observe quantised conductance. Further work on both the fabrication and measurement of nanowire devices may lead to their use as gates for ultra-short gate length HEMTs with improved frequency performance.

Fourthly, it would be interesting to investigate further the use of arbitrary structures pre-existing on a substrate for use as alignment markers. One of the principle advantages of using image processing is that the marker does not necessarily have to be strictly formed out of metal with good Z-contrast compared to the substrate or in simple geometric shapes. The flexibility offered by using image processing to perform marker searches for ebeam alignment opens up the possibility of being able to use all sorts of substrate features as alignment markers. There are cases where it would be useful to be able to align to structures already on a substrate such as carbon nanotubes or quantum dots. It would be interesting to investigate if the random placement of, for instance, carbon nanotubes could be used for alignment such that individual nanotubes could be contacted electrically in a precise fashion. By depositing a random array of quantum dots on a surface with carbon nanotubes it would be possible to use the Z-contrast of the dots as a map to indicate the position of the nanotubes. This or other techniques would be necessary if only BSE detectors were used for imaging.

There are also cases where it would be useful to be able to align to substrates where the deposition of markers is difficult or cannot be performed as the initial step. One example of this would be the use of anisotropically etched pits in Si as markers. The etching process produces a variation in the absolute position of the structures and therefore conventional metal markers would never be exactly at a known position relative to the

etched features. Making the marker in the same etch process removes some of this uncertainty but leaves "inverted pyramid" markers which have no sharp contrast with the background substrate and are hence difficult for conventional alignment techniques to locate. Anisotropic etching is used in the fabrication of atomic force microscopy (AFM) probes at the University of Glasgow, and an image processing technique has already been implemented to allow these etched pit markers to be located. Full details are provided in appendix B.

Lastly, there have recently been attempts with some success to measure the spot diameter and shape by using a deconvolution process [130, 131]. This works because when an object is imaged by an electron beam the image is effectively the convolution of the beam point spread function and the physical object being imaged. Therefore if the object is known it is theoretically possible to recover the beam point spread function by deconvolving the object from the acquired image. This is an ill-defined problem, but having an object pattern with a well defined peak in its autocorrelation aids the deconvolution process. For many of the same reasons that Penrose patterns are well suited to marker patterns they too may be well suited for spot size measurements via deconvolution. The traditional methods of measuring spot size uses a so called "knife-edge" [6, 132, 133], an approximate to an infinite contrast edge which reflects no beam on one side, and all the beam on the other and has no edge slope where the beam is being partly scattered and partly reflected to the detector. This has the limitation that the beam can only be scanned perpendicular to the edge and so the shape and size information obtained is similarly one-dimensional. With Penrose patterns the wide variety of edge directions would theoretically furnish a more complete measurement of the spot shape and size.

Appendices

A ImageJ Plugins

A.1 Correlate_Periodic.java

```
1  import ij.*;
   import ij.io.*;
   import ij.gui.*;
   import ij.plugin.*;
   import ij.process.*;
   import java.io.*;

   //Correlates an Image with a Reference displaying the resulting correlation.
   //Scans the reference across the image starting from the top-left
   //corner of image being overlapped by bottom-right of reference and ending
11 //with the bottom-right of the image overlapped by the top-left of
   //the reference. Uses periodic boundary conditions for the image and reference.
   //                  ____
   //                 |R   |
   //                 |   -|--          ____
   //                  --|-   |  ->   |I   |
   //                    |   I|  ->   |   -|--
   //                     ____          --|-   |
   //                                   |   R|
   //                                    ____
21

   public class Correlate_Periodic implements PlugIn {

     boolean debug = false;

     public void run(String arg){
       int k, ref_no, image_no, corrw, corrh, w, wi, h, hi, i, j, x, y, corrxmax,
           corrymax;
       float xoff, yoff, norm, corrmax, idatabkg, refdatabkg;
       ImagePlus refimp, iimp, corrimp, imp;
31     ImageProcessor refip, iip, corrip;
       float [] refdata, idata, itmp, corrdata;
       int[] wList;
       String[] titles;
       String ref_title, image_title;
       FileWriter outfile;
       SaveDialog sd;
       boolean cont;

       //Check that 1 or more images are open and adds their titles to variable "
           titles"
```

```
41      wList = WindowManager.getIDList();
        if (wList == null){
          IJ.noImage();
          return;
        }
        titles = new String[wList.length];
        for (k = 0; k < wList.length; k++){
          imp = WindowManager.getImage(wList[k]);
          if (imp != null)
            titles[k] = imp.getTitle();
51        else
            titles[k] = "";
        }

        //Reloads preferences from previous run
        ref_title = Prefs.get("Correlate_.ref_title", titles[0]);
        image_title =Prefs.get("Correlate_.image_title", titles[0]);
        ref_no = 0;
        image_no = 0;
        for (k = 0; k < wList.length; k++){
61        if(ref_title.equals(titles[k])){
            ref_no = k;
            break;
          }
        }
        for (k = 0; k < wList.length; k++){
          if(image_title.equals(titles[k])){
            image_no = k;
            break;
          }
71      }

        //Defines and shows dialog
        GenericDialog gd = new GenericDialog("Correlate", IJ.getInstance());
        gd.addChoice("Reference Mark", titles, titles[ref_no]);
        gd.addChoice("Image", titles, titles[image_no]);
        gd.showDialog();

        if (gd.wasCanceled()) return;

81      //Reads in selections from dialog
        refimp = WindowManager.getImage(wList[gd.getNextChoiceIndex()]);
        iimp = WindowManager.getImage(wList[gd.getNextChoiceIndex()]);

        //Sets preferences
        Prefs.set("Correlate_.ref_title", refimp.getTitle());
        Prefs.set("Correlate_.image_title", iimp.getTitle());

        refip = refimp.getProcessor();
        refip = refip.convertToFloat();
91      refdata = (float[]) refip.getPixels();
        w = refip.getWidth();
        h = refip.getHeight();
        norm = 0;
        for(i = 0; i < w*h; i++) {
          norm += refdata[i];
        }
        //IJ.write("w:"+IJ.d2s(w,0)+"  h:"+IJ.d2s(h,0));

        iip = iimp.getProcessor();
101     iip = iip.convertToFloat();
        itmp = (float[]) iip.getPixels();
```

```
wi = iip.getWidth();
hi = iip.getHeight();
//IJ.write("wi:"+IJ.d2s(wi,0)+"  hi:"+IJ.d2s(hi,0));

if(wi * hi != w * h){
  IJ.showMessage("Error", "Image and Reference Size do not match.");
  return;
}

//Make image array twice as long
idata = new float [wi*hi*2];
for(i = 0; i < (wi * hi); i++){
  idata[i] = itmp[i];
  idata[i + (wi*hi)] = itmp[i];
}

corrw = wi;
corrh = hi;
corrdata = new float [corrw*corrh];
//IJ.write("corrsize:"+IJ.d2s(corrsize,0));


for(i = 0; i < wi* hi; i++){
  //IJ.write("i:"+IJ.d2s(i,0));
  IJ.showStatus("Calculating... "+IJ.d2s(i,0)+"/"+IJ.d2s(wi*hi,0));
  for(j = 0; j < wi*hi; j++){
    //IJ.write("     j:"+IJ.d2s(j,0));
    //IJ.showStatus("Calculating... "+IJ.d2s(i,0)+"/"+IJ.d2s(corrw,0)+"  "+IJ.
        d2s(j,0)+"/"+IJ.d2s(corrh,0));
    corrdata[i] += idata[j + i] * refdata[j];
  }
  corrdata[i] /= norm;
}

corrip = new FloatProcessor(corrw, corrh);
corrip.setPixels(corrdata);
corrip.setMinAndMax(0,0);
corrimp = new ImagePlus("Correlation", corrip);
corrimp.show();

corrmax = corrdata[0];
corrxmax = 0;
corrymax = 0;

for(x = 0; x < corrw; x++) {
  for(y = 0; y < corrh; y++) {
    if(corrdata[x + (corrw * y)] > corrmax) {
      corrmax = corrdata[x + (corrw * y)];
      corrxmax = x;
      corrymax = y;
    }
  }
}

xoff = corrxmax;
yoff = corrymax;

IJ.write("xoff:"+IJ.d2s(xoff,4)+"     yoff:"+IJ.d2s(yoff,4)+"\n");

//Write offsets to a file
sd = new SaveDialog("Offsets File", "offsets", ".txt");
if(sd.getFileName() != null){
```

```
        try {
          outfile = new FileWriter(sd.getDirectory() + sd.getFileName());
          outfile.write("xoff\tyoff\n");
          outfile.write(IJ.d2s(xoff,4)+"\t"+IJ.d2s(yoff,4)+"\n");
          outfile.close();
        } catch (IOException ex) {
          ex.printStackTrace();
171     }
      }
      return;
    }     //End of Correlate_Periodic routine
}
```

A.2 Calculate_PCE.java

```
import ij.*;
import ij.gui.*;
import ij.plugin.*;
import ij.process.*;

/* Calculates the Peak to Correlation Energy ratio (PCE).

  Used as a figure of merit for autocorrelations of marker patterns
*/
public class Calculate_PCE implements PlugIn {

    public void run(String arg){
  int w,h, xc, yc;
  ImagePlus imp;
  ImageProcessor ip;
  float[] imagedata;
  float max, cval, Ey, avg, PCE, PCEc;

  //Check that 1 or more images are open and adds their titles to variable "titles
      "
  int[] wList = WindowManager.getIDList();
  if (wList == null)
      {
    IJ.noImage();
    return;
      }
  String[] titles = new String[wList.length];
  for (int i = 0; i < wList.length; i++)
      {
    imp = WindowManager.getImage(wList[i]);
    if (imp != null)
        titles[i] = imp.getTitle();
    else
        titles[i] = "";
      }

  //Reloads preferences from previous run
  String titleImage = Prefs.get("calculate_pce.title", titles[0]);
  int imageChoice = 0;
  for (int i = 0; i < wList.length; i++)
      {
    if(titleImage.equals(titles[i]))
        {
      imageChoice = i;
      break;
        }
      }

  //Defines and shows dialog
  GenericDialog gd = new GenericDialog("Calculate PCE", IJ.getInstance());
  gd.addChoice("Image", titles, titles[imageChoice]);
  gd.showDialog();

  if (gd.wasCanceled()) return;

  //Reads in selections from dialog
  imp = WindowManager.getImage(wList[gd.getNextChoiceIndex()]);

  //Sets preferences
  Prefs.set("calculate_pce.title", imp.getTitle());
```

```
    w = imp.getProcessor().getWidth();
    h = imp.getProcessor().getHeight();
    imagedata = new float[w*h];
    ImageProcessor ipi = imp.getProcessor();
65  ip = ipi.convertToFloat();
    imagedata = (float[])ip.getPixels();

    max = 0;
    for (int i=0; i<w*h; i++)
        {
      if(imagedata[i] > max) max = imagedata[i];
        }
    xc = (w-1)/2;
    yc = (h-1)/2;
75  cval = (float)ip.getPixelValue(xc,yc);
    Ey = 0;
    avg = 0;
    for (int i=0; i<w*h; i++)
        {
      Ey += sqr(imagedata[i]);
      avg += imagedata[i];
        }
    avg /= (w*h);
    PCE = sqr(max)/Ey;
85  PCEc = sqr(cval)/Ey;

    IJ.write("w : "+IJ.d2s((double)w,10)+
       "\th : "+IJ.d2s((double)h,10)+
       "\nxc : "+IJ.d2s((double)xc,10)+
       "\tyc : "+IJ.d2s((double)yc,10)+
       "\nMax : "+IJ.d2s((double)max,10)+
       "\nCentre = "+IJ.d2s((double)cval,10)+
       "\nAvg = "+IJ.d2s((double)avg,10)+
       "\nEy = "+IJ.d2s((double)Ey,10)+
95     "\nPCE = "+IJ.d2s((double)PCE,10)+
       "\nPCEc = "+IJ.d2s((double)PCEc,10)+"\n\n");

    return;
      }

      double sqr(double x) {return x*x;}

      float sqr(float x) {return x*x;}

105 }
```

A.3 PCE_Variation_Only_Ideal_Stack_Surface.java

```
import ij.*;
import ij.io.*;
import ij.gui.*;
import ij.plugin.*;
import ij.process.*;
import java.io.*;
import ij.plugin.filter.*;
import java.awt.*;
import java.awt.image.*;
import java.math.*;
import java.util.*;
import ij.measure.*;
//Allows the variations of the PCE measure to be investigated

public class PCE_Variation_Only_Ideal_Stack_Surface implements PlugIn {

  int debug = 0;   //1 for debug messages from gaussian kernel, 2 for convolve, 3
      for full_process, 4 for main prog
  int first, i, totalruns = 0;
  boolean showimages = false;
  ImageStack samBRfullis, CCis;

  //global variables for surface plotter subroutine
  static int plotWidth = 350;
  static int polygonMultiplier = 200;
  static boolean oneToOne = false;
  static boolean firstTime = true;
  static boolean showWireframe=true;
  static boolean showGrayscale=false;
  static boolean showAxis=true;
  static boolean whiteBackground=false;
  static boolean blackFill=false;
  static boolean smooth = false;
  ImagePlus img;
  int[] x,y;
  boolean invertedLut;
  double angleInDegrees = 35;
  double angle = (angleInDegrees/360.0)*2.0*Math.PI;
  double angle2InDegrees = 15.0;
  double angle2 = (angle2InDegrees/360.0)*2.0*Math.PI;
  double yinc2 = Math.sin(angle2);
  double p1x, p1y;  // left bottom corner
  double p2x, p2y;  // center bottom corner
  double p3x, p3y;  // right  bottom corner
  LookUpTable lut;
  //end of surface plotter global variables

  public void run(String arg){
    int x, imageChoice, rmin, rmax, rstp, r, rruns, samintmin, samintmax,
        samintstp, samint, samintruns, offsetvar, xoffmin, xoffmax, xoffstp,
        xoffruns, yoffmin, yoffmax, yoffstp, yoffruns, offruns, xoff, yoff,
        stdevnoisemin, stdevnoisemax, stdevnoisestp, stdevnoise, stdevnoiseruns,
        angrotruns, w, h;
    ImagePlus imp, samBRfullstackimp, CCstackimp, CCsurfaceplotimp;
    ImageStack CCsurfaceplotis;
    double angrotmin, angrotmax, angrotstp, angrot;
    boolean alloffsets, showimages;
    int[] wList;
    String[] titles, offchoices;
    String titleImage, path;
```

```
        float[][] results;
        float[] pces;
        FileWriter outfile;
        SaveDialog sd;
        DirectoryChooser dc;

        //Check that 1 or more images are open and adds their titles to variable "
            titles"
        wList = WindowManager.getIDList();
        if (wList == null){
65        IJ.noImage();
          return;
        }
        titles = new String[wList.length];
        for (i = 0; i < wList.length; i++){
          imp = WindowManager.getImage(wList[i]);
          if (imp != null)
            titles[i] = imp.getTitle();
          else
            titles[i] = "";
75      }

        offchoices = new String[4];
        offchoices[0] = "Single Offset Only";
        offchoices[1] = "Center Offset Only";
        offchoices[2] = "Range of Offsets";
        offchoices[3] = "All Valid Offsets";

        //Reloads preferences from previous run
        titleImage = Prefs.get("PCE_Variation.title", titles[0]);
85      imageChoice = 0;
        for (i = 0; i < wList.length; i++){
          if(titleImage.equals(titles[i])){
            imageChoice = i;
            break;
          }
        }

        rmin = (int)Prefs.get("PCE_Variation.rmin", 0);
        rmax = (int)Prefs.get("PCE_Variation.rmax", 0);
95      rstp = (int)Prefs.get("PCE_Variation.rstp", 0);
        samintmin = (int)Prefs.get("PCE_Variation.samintmin", 1);
        samintmax = (int)Prefs.get("PCE_Variation.samintmax", 1);
        samintstp = (int)Prefs.get("PCE_Variation.samintstp", 0);
        offsetvar = (int)Prefs.get("PCE_Variation.offsetvar", 1);
        stdevnoisemin = (int)Prefs.get("PCE_Variation.stdevnoisemin", 0);
        stdevnoisemax = (int)Prefs.get("PCE_Variation.stdevnoisemax", 0);
        stdevnoisestp = (int)Prefs.get("PCE_Variation.stdevnoisestp", 0);
        angrotmin = (double)Prefs.get("PCE_Variation.angrotmin",0);
        angrotmax = (double)Prefs.get("PCE_Variation.angrotmax",0);
105     angrotstp = (double)Prefs.get("PCE_Variation.angrotstp",0);
        showimages = (boolean)Prefs.get("PCE_Variation.showimages", false);

        //Defines and shows dialog
        GenericDialog gd = new GenericDialog("PCE Variation", IJ.getInstance());
        gd.addChoice("Image", titles, titles[imageChoice]);
        gd.addNumericField("Radius min", rmin, 0);
        gd.addNumericField("Radius max", rmax, 0);
        gd.addNumericField("Radius step", rstp, 0);
        gd.addNumericField("Sampling Interval min", samintmin, 0);
115     gd.addNumericField("Sampling Interval max", samintmax, 0);
        gd.addNumericField("Sampling Interval step", samintstp, 0);
```

```
        gd.addChoice("Offset Variation", offchoices, offchoices[offsetvar]);
        gd.addNumericField("St Dev of Noise min", stdevnoisemin, 1);
        gd.addNumericField("St Dev of Noise max", stdevnoisemax, 1);
        gd.addNumericField("St Dev of Noise step", stdevnoisestp, 1);
        gd.addNumericField("Rotation Angle min", angrotmin, 3);
        gd.addNumericField("Rotation Angle max", angrotmax, 3);
        gd.addNumericField("Rotation Angle step", angrotstp, 3);
        gd.addCheckbox("Show Intermediate Images", showimages);
125     gd.showDialog();

        if (gd.wasCanceled()) return;

        //Reads in selections from dialog
        imp = WindowManager.getImage(wList[gd.getNextChoiceIndex()]);
        rmin = (int)gd.getNextNumber();
        rmax = (int)gd.getNextNumber();
        rstp  = (int)gd.getNextNumber();
        samintmin = (int)gd.getNextNumber();
135     samintmax = (int)gd.getNextNumber();
        samintstp = (int)gd.getNextNumber();
        offsetvar = gd.getNextChoiceIndex();
        stdevnoisemin = (int)gd.getNextNumber();
        stdevnoisemax = (int)gd.getNextNumber();
        stdevnoisestp = (int)gd.getNextNumber();
        angrotmin = (double)gd.getNextNumber();
        angrotmax = (double)gd.getNextNumber();
        angrotstp = (double)gd.getNextNumber();
        showimages = gd.getNextBoolean();
145
        //Sets preferences
        Prefs.set("PCE_Variation.title", imp.getTitle());
        Prefs.set("PCE_Variation.rmin", rmin);
        Prefs.set("PCE_Variation.rmax", rmax);
        Prefs.set("PCE_Variation.rstp", rstp);
        Prefs.set("PCE_Variation.samintmin", samintmin);
        Prefs.set("PCE_Variation.samintmax", samintmax);
        Prefs.set("PCE_Variation.samintstp", samintstp);
        Prefs.set("PCE_Variation.offsetvar", offsetvar);
155     Prefs.set("PCE_Variation.stdevnoisemin", stdevnoisemin);
        Prefs.set("PCE_Variation.stdevnoisemax", stdevnoisemax);
        Prefs.set("PCE_Variation.stdevnoisestp", stdevnoisestp);
        Prefs.set("PCE_Variation.angrotmin", angrotmin);
        Prefs.set("PCE_Variation.angrotmax", angrotmax);
        Prefs.set("PCE_Variation.angrotstp", angrotstp);
        Prefs.set("PCE_Variation.showimages", showimages);

        xoffmin = 0;
        xoffmax = 0;
165     xoffstp = 0;
        xoff = 0;
        yoffmin = 0;
        yoffmax = 0;
        yoffstp = 0;
        yoff = 0;

        if (offsetvar==0) {  //Single Offset
          GenericDialog gd1 = new GenericDialog("Single Offset Selection", IJ.
              getInstance());
          gd1.addNumericField("X Offset", xoff, 0);
175       gd1.addNumericField("Y Offset", yoff, 0);
          gd1.showDialog();
```

```
      xoffmin = (int)gd1.getNextNumber();
      yoffmin = (int)gd1.getNextNumber();

      offruns = 1;
    } else if (offsetvar==1) {  //Center offset
      offruns = 1;
    } else if (offsetvar==2) {  //Range of offsets
      GenericDialog gd1 = new GenericDialog("Offset Range Selection", IJ.
          getInstance());
      gd1.addNumericField("X Offset Min", xoffmin, 0);
      gd1.addNumericField("X Offset Max", xoffmax, 0);
      gd1.addNumericField("X Offset Step", xoffstp, 0);
      gd1.addNumericField("Y Offset Min", yoffmin, 0);
      gd1.addNumericField("Y Offset Max", yoffmax, 0);
      gd1.addNumericField("Y Offset Step", yoffstp, 0);
      gd1.showDialog();

      xoffmin = (int)gd1.getNextNumber();
      xoffmax = (int)gd1.getNextNumber();
      xoffstp = (int)gd1.getNextNumber();
      yoffmin = (int)gd1.getNextNumber();
      yoffmax = (int)gd1.getNextNumber();
      yoffstp = (int)gd1.getNextNumber();

      if(xoffstp == 0) {
        xoffruns = 1;
      } else {
        xoffruns = (int)Math.floor((xoffmax-xoffmin)/xoffstp) + 1;
      }
      if(yoffstp == 0) {
        yoffruns = 1;
      } else {
        yoffruns = (int)Math.floor((yoffmax-yoffmin)/yoffstp) + 1;
      }
      offruns = xoffruns * yoffruns;
    } else if (offsetvar==3) {  //All offsets
      samint = samintmin;
      xoffstp = 1;
      yoffstp = 1;
      offruns = -1;
      do {
        offruns += sqr(samint);
        samint += samintstp;
      } while(samintstp!=0 && samint<=samintmax);
    } else return;

    if(rstp == 0) {
      rruns = 1;
    } else {
      rruns = (int)Math.floor((rmax-rmin)/rstp) + 1;
    }
    if(samintstp == 0 || offsetvar > 2) {
      samintruns = 1;
    } else {
      samintruns = (int)Math.floor((samintmax-samintmin)/samintstp) + 1;
    }
    if(stdevnoisestp == 0) {
      stdevnoiseruns = 1;
    } else {
      stdevnoiseruns = (int)Math.floor((stdevnoisemax-stdevnoisemin)/stdevnoisestp
          ) + 1;
    }
```

```
    if(angrotstp == 0) {
      angrotruns = 1;
    } else {
      angrotruns = (int)Math.floor((angrotmax-angrotmin)/angrotstp) + 1;
    }
    totalruns = rruns * samintruns * stdevnoiseruns * angrotruns * offruns + 1;
    results = new float[totalruns][11];

    if(debug==4) {
      IJ.write("rmin\trmax\trstep\trruns\n"+IJ.d2s(rmin,0)+"\t"+IJ.d2s(rmax,0)+"\t
          "+IJ.d2s(rstp,0)+"\t"+IJ.d2s(rruns,0));
      IJ.write("samintmin\tsamintmax\tsamintstep\tsamintruns\n"+IJ.d2s(samintmin
          ,0)+"\t"+IJ.d2s(samintmax,0)+"\t"+IJ.d2s(samintstp,0)+"\t"+IJ.d2s(
          samintruns,0));
      IJ.write("stdevnoisemin\tstdevnoisemax\tstdevnoisestep\tstdevnoiseruns\n"+IJ
          .d2s(stdevnoisemin,0)+"\t"+IJ.d2s(stdevnoisemax,0)+"\t"+IJ.d2s(
          stdevnoisestp,0)+"\t"+IJ.d2s(stdevnoiseruns,0));
      IJ.write("angrotmin\tangrotmax\tangrotstep\tangrotruns\n"+IJ.d2s(angrotmin
          ,0)+"\t"+IJ.d2s(angrotmax,0)+"\t"+IJ.d2s(angrotstp,0)+"\t"+IJ.d2s(
          angrotruns,0));
      IJ.write("Total runs = " + IJ.d2s(totalruns,0));
    }

    dc = new DirectoryChooser("Choose where to save image files");
    path = dc.getDirectory();

    pces = new float[5];
    i=0;
    r = rmin;
    do {
      samint = samintmin;
      do {
        first = 0;
        if (offsetvar == 1) {
          xoffmin = (int)Math.floor((samint - 1)/2);
          yoffmin = (int)Math.floor((samint - 1)/2);
        }
        if (offsetvar == 3) {
          xoffmax = samint - 1;
          yoffmax = samint - 1;
        }
        xoff = xoffmin;
        do {
          yoff = yoffmin;
          do {
            stdevnoise = stdevnoisemin;
            do {
              angrot = angrotmin;
              do {
                dofull_process(imp, r, samint, xoff, yoff, stdevnoise, angrot,
                    showimages, pces);
                if(debug==4) IJ.write("r : " +IJ.d2s(r,1)+"\nsamint : "+IJ.d2s(
                    samint,1)+"\nxoff : "+IJ.d2s(xoff,1)+"\nyoff : "+IJ.d2s(yoff
                    ,1)+"\nstdevnoise : "+IJ.d2s(stdevnoise,1)+"\nangrot : "+IJ.
                    d2s(angrot,1)+"\ni : "+IJ.d2s(i,1)+"\npce1 : "+IJ.d2s(pces
                    [2],8));
                results[i][0] = r;
                results[i][1] = samint;
                results[i][2] = xoff;
                results[i][3] = yoff;
                results[i][4] = stdevnoise;
                results[i][5] = (float)angrot;
```

```
                for(x = 0; x < 5; x++) {
                  results[i][x+6]=pces[x];
                }
                i++;
                first++;
                angrot += angrotstp;
                IJ.showProgress((double) i/totalruns);
              } while(angrotstp!=0 && angrot<=angrotmax);
              stdevnoise += stdevnoisestp;
            } while(stdevnoisestp!=0 && stdevnoise<=stdevnoisemax);
            yoff += yoffstp;
          } while(yoffstp!=0 && yoff<=yoffmax);
          xoff += xoffstp;
        } while(xoffstp!=0 && xoff<=xoffmax);

        samBRfullstackimp = new ImagePlus("Sampled Images S"+IJ.d2s(samint,0),
            samBRfullis);
        samBRfullstackimp.show();

        CCstackimp = new ImagePlus("Crosscorrelations with Ideal S"+IJ.d2s(samint
            ,0), CCis);
        CCstackimp.show();

        lut = CCstackimp.createLut();
        img = CCstackimp;
        CCsurfaceplotis = dosurfaceplot(CCis);
        CCsurfaceplotimp = new ImagePlus("Surface Plot - Crosscorrelation with
            Ideal S"+IJ.d2s(samint,0), CCsurfaceplotis);
        CCsurfaceplotimp.show();
        if(samBRfullstackimp.getImageStackSize() > 1) {
          new FileSaver(samBRfullstackimp).saveAsTiffStack(path +
              samBRfullstackimp.getTitle() + ".tif");
        } else {
          new FileSaver(samBRfullstackimp).saveAsTiff(path + samBRfullstackimp.
              getTitle() + ".tif");
        }
        samBRfullstackimp.close();
        if(samBRfullstackimp.getImageStackSize() > 1) {
          new FileSaver(CCstackimp).saveAsTiffStack(path + CCstackimp.getTitle() +
               ".tif");
        } else {
          new FileSaver(CCstackimp).saveAsTiff(path + CCstackimp.getTitle() + ".
              tif");
        }
        CCstackimp.close();
        if(samBRfullstackimp.getImageStackSize() > 1) {
          new FileSaver(CCsurfaceplotimp).saveAsTiffStack(path + CCsurfaceplotimp.
              getTitle() + ".tif");
        } else {
          new FileSaver(CCsurfaceplotimp).saveAsTiff(path + CCsurfaceplotimp.
              getTitle() + ".tif");
        }
        CCsurfaceplotimp.close();

        samint += samintstp;
      } while(samintstp!=0 && samint<=samintmax);
      r += rstp;
    } while(rstp!=0 && r<=rmax);

    sd = new SaveDialog("Results File", "results", ".txt");
    outfile = null;
    try {
```

```
        outfile = new FileWriter(sd.getDirectory() + sd.getFileName());

        outfile.write("r\tsamint\txoff\tyoff\tstdevnoise\tangrot\tCCidealMax\
            tCCidealcval\tCCidealEy\tCCidealPCE\tCCidealPCEc\n");

345     for(i = 0; i < totalruns; i++) {
          for(x = 0; x < 11; x++) {
            outfile.write(IJ.d2s(results[i][x],8)+"\t");
          }
          outfile.write("\n");
        }
        outfile.close();
      } catch (IOException ex) {
        ex.printStackTrace();
      }
355
      return;
    }     //End of PCE_Variation routine


          //////////////
          //          //
          //   SUBS   //
          //          //
          //////////////
365

    public void dofull_process(ImagePlus imp, int gbrad, int samint, int xoff, int
        yoff, int stdevnoise, double rotang, boolean showimages, float[] result){
      //int debug = 0;   //1 for debug messages from gaussian kernel, 2 for convolve
          , 3 for main prog

      //------Inputs------
      //ImagePlus imp;
      //int gbrad, samint, xoff, yoff, stdevnoise;
      //double rotang;
      //boolean showimages;
375
      //------Internal Variables------
      int radius, size, w, wi, h, hi, xc, yc, xoc, yoc, shftx, shfty;
      double rotangRadians;
      ImagePlus kernelimp, blurredimp, rotimp, samBRimp, samBimp, samOimp,
          samBRfullimp, flippedBRimp, flippedBimp, flippedsamOimp, flippedOimp;
      ImageProcessor ip, kernelip, blurredip, rotip, samBRip, samBip, samOip,
          samBRfullip, flippedBRip, flippedBip, flippedsamOip, flippedOip;
      float[] imagedata, kerneldata, blurreddata, rotdata, samBRdata, samBdata,
          samOdata, samBRfulldata, flippedBRdata, flippedBdata, flippedsamOdata,
          flippedOdata;

      //------Outputs------
      ImagePlus ACimp, CCunrotimp, CCsamidealimp, CCidealimp;
385   ImageProcessor ACip, CCunrotip, CCsamidealip, CCidealip;
      float [] ACdata, CCunrotdata, CCsamidealdata, CCidealdata;
      float ACMax, ACcval, ACEy, ACPCE, ACPCEc, CCunrotMax, CCunrotcval, CCunrotEy,
          CCunrotPCE, CCunrotPCEc, CCsamidealMax, CCsamidealcval, CCsamidealEy,
          CCsamidealPCE, CCsamidealPCEc, CCidealMax, CCidealcval, CCidealEy,
          CCidealPCE, CCidealPCEc;
      //float [] result;

      if(debug==3) IJ.write("r : " +IJ.d2s(gbrad,1)+"\nsamint : "+IJ.d2s(samint,1)+"
          \nstdevnoise : "+IJ.d2s(stdevnoise,1)+"\nangrot : "+IJ.d2s(rotang,1));
```

```
    //-----------------------------------------------------//
    //Convolve base image with a gaussian of radius gbrad//
    //-----------------------------------------------------//
395   ip = imp.getProcessor();
      ip = ip.convertToFloat();

      imagedata = (float[])ip.getPixels();
      wi = imp.getProcessor().getWidth();
      hi = imp.getProcessor().getHeight();
      blurreddata = new float[wi*hi];

      radius = gbrad + 1;
      if(radius > 1) {
405     kerneldata = makeKernel(radius);
        size = radius*2-1;
        kernelip = new FloatProcessor(size, size);
        kernelip.setPixels(kernedata);
        kernelip.setMinAndMax(0,0);
        kernelimp = new ImagePlus("Gaussian Blur Kernel",kernelip);
        if (showimages) kernelimp.show();
        doConvolve(imp, kernelimp, blurreddata);
      } else {
        blurreddata = (float[])imagedata;
415   }
      blurredip = new FloatProcessor(wi,hi);
      blurredip.setPixels(blurreddata);
      blurredip.setMinAndMax(0,0);
      blurredimp = new ImagePlus("Blurred",blurredip);
      if (showimages) blurredimp.show();

    //---------------------------------//
    //Rotated by angle given by rotang//
    //---------------------------------//
425   if(rotang > 0) {
        if(rotang>180) rotang -= 180;
        if(rotang<=90) {
          rotangRadians = Math.toRadians(rotang);
          w = (int) Math.ceil(wi*Math.cos(rotangRadians) + hi*Math.sin(rotangRadians
              ));
          h = (int) Math.ceil(hi*Math.cos(rotangRadians) + wi*Math.sin(rotangRadians
              ));
        } else {
          rotangRadians = Math.toRadians(rotang - 90);
          w = (int) Math.ceil(hi*Math.cos(rotangRadians) + wi*Math.sin(rotangRadians
              ));
          h = (int) Math.ceil(wi*Math.cos(rotangRadians) + hi*Math.sin(rotangRadians
              ));
435     }
        if(debug==3) IJ.write("Before oddifying\nw = "+IJ.d2s(w,3)+"\th = "+IJ.d2s(h
            ,3));
        int tempw = (int) Math.floor(w/2);
        int temph = (int) Math.floor(h/2);
        if (w - 2*tempw == 0) w += 1;
        if (h - 2*temph == 0) h += 1;
        if(debug==3) IJ.write("After oddifying\nw = "+IJ.d2s(w,3)+"\th = "+IJ.d2s(h
            ,3));
        xc = (w - wi)/2;
        yc = (h - hi)/2;
        if(debug==3) IJ.write("Rotang degrees "+IJ.d2s(rotang,4)+"\nRotang radians "
            +IJ.d2s(rotangRadians,4)+"\nwi,hi "+IJ.d2s((double)wi,0)+","+IJ.d2s((
            double)hi,0)+"\nw,h "+IJ.d2s((double)w,0)+","+IJ.d2s((double)h,0));
445     rotip = blurredip.createProcessor(w, h);
```

```
        rotip.setValue(0.0);
        rotip.fill();
        rotip.insert(blurredip, xc, yc);
        rotip.setInterpolate(false);
        rotip.rotate(rotang);
      } else {
        w = wi;
        h = hi;
        rotip = blurredip.duplicate();
455   }
      rotdata = (float[]) rotip.getPixels();
      rotip.setMinAndMax(0,0);
      rotimp = new ImagePlus("Rotated",rotip);
      if (showimages) rotimp.show();

    /*//-------------------------------------------------------------------------//
      //Sample blurred, rotated image at spacing samint, starting from (xoff, yoff)//
      //-------------------------------------------------------------------------//
      int wo = (int) Math.ceil((double)w / samint);
465   int ho = (int) Math.ceil((double)h / samint);
      if(debug==3) IJ.write("wo : " + wo + "\tho : " +ho);

      samBRdata = new float[wo*ho];

      if(samint>1){
        for(int x=xoff; x<w; x+=samint){
          for(int y=yoff; y<h; y+=samint){
            if(debug==3) IJ.write("x : " + x + "\ty : " + y + "\tDi : " + IJ.d2s((
                double)rotdata[x+w*y]));
            samBRdata[(x-xoff)/samint+wo*(y-yoff)/samint]=rotdata[x+w*y];
475       }
        }
      } else {
        samBRdata = rotdata;
      }
      samBRip = new FloatProcessor(wo,ho);
      samBRip.setPixels(samBRdata);
      samBRip.setMinAndMax(0,0);
      samBRimp = new ImagePlus("Blurred Rotated Sampled",samBRip);
      if (showimages) samBRimp.show();
485
    //-------------------------------------------------------------------------//

    //Sample blurred, UNrotated image at spacing samint, starting from (xoff, yoff)
        //
    //-------------------------------------------------------------------------//

      int wo2 = (int) Math.ceil((double)wi / samint);
      int ho2 = (int) Math.ceil((double)hi / samint);
      if(debug==3) IJ.write("wo : " + wo + "\tho : " +ho);

      samBdata = new float[wo2*ho2];
      if(samint>1){
495     for(int x=xoff; x<wi; x+=samint){
          for(int y=yoff; y<hi; y+=samint){
            if(debug==3) IJ.write("x : " + x + "\ty : " + y + "\tDi : " + IJ.d2s((
                double)blurreddata[x+wi*y]));
            samBdata[(x-xoff)/samint+wo2*(y-yoff)/samint]=blurreddata[x+wi*y];
          }
        }
      } else {
        samBdata = blurreddata;
```

```
        }
        samBip = new FloatProcessor(wo2,ho2);
505     samBip.setPixels(samBdata);
        samBip.setMinAndMax(0,0);
        samBimp = new ImagePlus("Blurred Sampled",samBip);
        if (showimages) samBimp.show();

      //-------------------------------------------------------------------//
      //Sample original image at spacing samint, starting from (xoff, yoff)//
      //-------------------------------------------------------------------//
        samOdata = new float[wo2*ho2];
        if(samint>1){
515       for(int x=xoff; x<wi; x+=samint){
            for(int y=yoff; y<hi; y+=samint){
              if(debug==3) IJ.write("x : " + x + "\ty : " + y + "\tDi : " + IJ.d2s((
                  double)imagedata[x+wi*y]));
              samOdata[(x-xoff)/samint+wo2*(y-yoff)/samint]=imagedata[x+wi*y];
            }
          }
        } else {
          samOdata = imagedata;
        }
        samOip = new FloatProcessor(wo2,ho2);
525     samOip.setPixels(samOdata);
        samOip.setMinAndMax(0,0);
        samOimp = new ImagePlus("Original Sampled",samOip);
        if (showimages) samOimp.show();
        */
      //-------------------------------------------------------------------//
      //Produce image of the blurred rotated image with same pixel size as original//
      //but with data only at the sampled points and zeros inbetween      //
      //-------------------------------------------------------------------//
        samBRfulldata = new float[w*h];
535     if(samint>1){
          for(int x=0; x<w; x++){
            for(int y=0; y<h; y++){
              float temp1 = (float)(x-xoff)/samint;
              float temp2 = (float)Math.ceil(temp1);
              float temp3 = (float)(y-yoff)/samint;
              float temp4 = (float)Math.ceil(temp3);
              if(temp1==temp2&&temp3==temp4){
                int xsam = x - xoff;
                int ysam = y - yoff;
545             samBRfulldata[x+w*y] = rotdata[x+w*y];
              }
            }
          }
        } else {
          samBRfulldata = rotdata;
        }
        samBRfullip = new FloatProcessor(w,h);
        samBRfullip.setPixels(samBRfulldata);
        samBRfullip.setMinAndMax(0,0);
555     samBRfullimp = new ImagePlus("Blurred Rotated Sampled at Original Dimensions",
            samBRfullip);
        if (showimages) samBRfullimp.show();
        //samBRfullimp.show();

        if(first==0) samBRfullis = new ImageStack(w,h);
        samBRfullis.addSlice("S: " + IJ.d2s((double)samint,0) + "Offset: " +IJ.d2s((
            double)xoff,0) + "," + IJ.d2s((double)yoff,0), samBRfullip);
```

```
    //------------//
    //Add in AWGN//
    //------------//
    /*     noiseBRdata = new float[wo*ho];
      for (int i=0; i<samdata.length; i++) noisedata[i] = samdata[i];
      noiseip = new FloatProcessor(wo,ho);
      noiseip.setPixels(noisedata);
      noiseip.noise(stdevnoise);
      noiseimp = new ImagePlus("Noisy", noiseip);
      if (showimages) noiseimp.show();
    */

  /*//----------------------//
    //Create Flipped Images//
    //----------------------//
      flippedBRdata = new float[wo*ho];
      for (int i=0; i<wo*ho; i++) flippedBRdata[i] = samBRdata[wo*ho-1-i];
      flippedBRip = new FloatProcessor(wo,ho);
      flippedBRip.setPixels(flippedBRdata);
      flippedBRip.setMinAndMax(0,0);
      flippedBRimp = new ImagePlus("Flipped BR", flippedBRip);
      if (showimages) flippedBRimp.show();

      flippedBdata = new float[wo2*ho2];
      for (int i=0; i<wo2*ho2; i++) flippedBdata[i] = samBdata[wo2*ho2-1-i];
      flippedBip = new FloatProcessor(wo2,ho2);
      flippedBip.setPixels(flippedBdata);
      flippedBip.setMinAndMax(0,0);
      flippedBimp = new ImagePlus("Flipped B", flippedBip);
      if (showimages) flippedBimp.show();

      flippedsamOdata = new float[wo2*ho2];
      for (int i=0; i<wo2*ho2; i++) flippedsamOdata[i] = samOdata[wo2*ho2-1-i];
      flippedsamOip = new FloatProcessor(wo2,ho2);
      flippedsamOip.setPixels(flippedsamOdata);
      flippedsamOip.setMinAndMax(0,0);
      flippedsamOimp = new ImagePlus("Flipped Sampled O", flippedsamOip);
      if (showimages) flippedsamOimp.show();
      */
      flippedOdata = new float[wi*hi];
      for (int i=0; i<wi*hi; i++) flippedOdata[i] = imagedata[wi*hi-1-i];
      flippedOip = new FloatProcessor(wi,hi);
      flippedOip.setPixels(flippedOdata);
      flippedOip.setMinAndMax(0,0);
      flippedOimp = new ImagePlus("Flipped Original", flippedOip);
      if (showimages) flippedOimp.show();
      int index = 0;
  /*//-------------------------------------------------//
    //Correlations - Autocorrelation Sampled B & R//
    //-------------------------------------------------//
      ACdata = new float[wo*ho];
      doConvolve(samBRimp, flippedBRimp, ACdata);
      ImageProcessor ACbip = new FloatProcessor(wo,ho);
      ACbip.setPixels(ACdata);

      shftx = (wo2 - wo)/2;
      shfty = (ho2 - ho)/2;
      ACip = ACbip.createProcessor(wo2, ho2);
      ACip.setValue(0.0);
      ACip.fill();
      ACip.insert(ACbip, shftx, shfty);
      ACip.setMinAndMax(0,0);
```

```
        ACimp = new ImagePlus("Autocorrelation", ACip);
625     if (showimages) ACimp.show();

        ACMax = (float)ACip.getMax();
        xoc = (wo2-1)/2;
        yoc = (ho2-1)/2;
        ACcval = (float)ACip.getPixelValue(xoc,yoc);
        ACEy = 0;
        for (int i=0; i<wo2*ho2; i++) ACEy += sqr(ACdata[i]);
        ACPCE = sqr(ACMax)/ACEy;
        ACPCEc = sqr(ACcval)/ACEy;
635     //IJ.write("Autocorrelate\n"+IJ.d2s((double)ACMax,4)+"\n"+IJ.d2s((double)sqr(
            ACMax),4)+"\n"+IJ.d2s((double)ACEy,4)+"\n"+IJ.d2s((double)ACPCE,7));
        result[index++] = ACMax;
        result[index++] = ACcval;
        result[index++] = ACEy;
        result[index++] = ACPCE;
        result[index++] = ACPCEc;

      //-------------------------------------------------------//
      //Correlations - Crosscorrelation with unrotated//
      //-------------------------------------------------------//
645     CCunrotdata = new float[wo*ho];
        doConvolve(samBRimp, flippedBimp, CCunrotdata);
        ImageProcessor CCunrotbip = new FloatProcessor(wo,ho);
        CCunrotbip.setPixels(CCunrotdata);

        CCunrotip = CCunrotbip.createProcessor(wo2, ho2);
        CCunrotip.setValue(0.0);
        CCunrotip.fill();
        CCunrotip.insert(CCunrotbip, shftx, shfty);
        CCunrotip.setMinAndMax(0,0);
655     CCunrotimp = new ImagePlus("Crosscorrelation with unrotated", CCunrotip);
        if (showimages) CCunrotimp.show();

        CCunrotMax = (float)CCunrotip.getMax();
        CCunrotcval = (float)CCunrotip.getPixelValue(xoc,yoc);
        CCunrotEy = 0;
        for (int i=0; i<wo2*ho2; i++) CCunrotEy += sqr(CCunrotdata[i]);
        CCunrotPCE = sqr(CCunrotMax)/CCunrotEy;
        CCunrotPCEc = sqr(CCunrotcval)/CCunrotEy;
        //IJ.write("Crosscorrelate unrotated\n"+IJ.d2s((double)CCunrotMax,4)+"\n"+IJ.
            d2s((double)sqr(CCunrotMax),4)+"\n"+IJ.d2s((double)CCunrotEy,4)+"\n"+IJ.
            d2s((double)CCunrotPCE,7));
665     result[index++] = CCunrotMax;
        result[index++] = CCunrotcval;
        result[index++] = CCunrotEy;
        result[index++] = CCunrotPCE;
        result[index++] = CCunrotPCEc;

      //-------------------------------------------------------//
      //Correlations - Crosscorrelation with sampled ideal//
      //-------------------------------------------------------//
        CCsamidealdata = new float[wo*ho];
675     doConvolve(samBRimp, flippedsamOimp, CCsamidealdata);
        ImageProcessor CCsamidealbip = new FloatProcessor(wo,ho);
        CCsamidealbip.setPixels(CCsamidealdata);

        CCsamidealip = CCsamidealbip.createProcessor(wo2, ho2);
        CCsamidealip.setValue(0.0);
        CCsamidealip.fill();
        CCsamidealip.insert(CCsamidealbip, shftx, shfty);
```

```
        CCsamidealip.setMinAndMax(0,0);
        CCsamidealimp = new ImagePlus("Crosscorrelation with Sampled Ideal",
            CCsamidealip);
685     if (showimages) CCsamidealimp.show();

        CCsamidealMax = (float)CCsamidealip.getMax();
        CCsamidealcval = (float)CCsamidealip.getPixelValue(xoc,yoc);
        CCsamidealEy = 0;
        for (int i=0; i<wo*ho; i++) CCsamidealEy += sqr(CCsamidealdata[i]);
        CCsamidealPCE = sqr(CCsamidealMax)/CCsamidealEy;
        CCsamidealPCEc = sqr(CCsamidealcval)/CCsamidealEy;
        //IJ.write("Crosscorrelate sampled ideal\n"+IJ.d2s((double)CCsamidealMax,4)+"\
            n"+IJ.d2s((double)sqr(CCsamidealMax),4)+"\n"+IJ.d2s((double)CCsamidealEy
            ,4)+"\n"+IJ.d2s((double)CCsamidealPCE,7));
        result[index++] = CCsamidealMax;
695     result[index++] = CCsamidealcval;
        result[index++] = CCsamidealEy;
        result[index++] = CCsamidealPCE;
        result[index++] = CCsamidealPCEc;
        */
      //-----------------------------------------//
      //Correlations - Crosscorrelation with ideal//
      //-----------------------------------------//
        CCidealdata = new float[w*h];
        doConvolve(samBRfullimp, flippedOimp, CCidealdata);
705     ImageProcessor CCidealbip = new FloatProcessor(w,h);
        CCidealbip.setPixels(CCidealdata);

        shftx = (wi - w)/2;
        shfty = (hi - h)/2;
        CCidealip = CCidealbip.createProcessor(wi, hi);
        CCidealip.setValue(0.0);
        CCidealip.fill();
        CCidealip.insert(CCidealbip, shftx, shfty);
        CCidealip.setMinAndMax(0,0);
715     CCidealimp = new ImagePlus("Crosscorrelation with ideal", CCidealip);
        if (showimages) CCidealimp.show();
        //CCidealimp.show();

        if(first==0) CCis = new ImageStack(wi,hi);
        CCis.addSlice("S: " + IJ.d2s((double)samint,0) + "Offset: " +IJ.d2s((double)
            xoff,0) + "," + IJ.d2s((double)yoff,0), CCidealip);

        CCidealMax = (float)CCidealip.getMax();
        xc = (wi-1)/2;
        yc = (hi-1)/2;
725     CCidealcval = (float)CCidealip.getPixelValue(xc,yc);
        CCidealEy = 0;
        for (int i=0; i<wi*hi; i++) CCidealEy += sqr(CCidealdata[i]);
        CCidealPCE = sqr(CCidealMax)/CCidealEy;
        CCidealPCEc = sqr(CCidealcval)/CCidealEy;
        //IJ.write("Crosscorrelate with ideal\n"+IJ.d2s((double)CCidealMax,4)+"\n"+IJ.
            d2s((double)sqr(CCidealMax),4)+"\n"+IJ.d2s((double)CCidealEy,4)+"\n"+IJ.
            d2s((double)CCidealPCE,7));
        result[index++] = CCidealMax;
        result[index++] = CCidealcval;
        result[index++] = CCidealEy;
        result[index++] = CCidealPCE;
735     result[index++] = CCidealPCEc;

      //-----------------------------------//
      //Combine results into single array//
```

```
    //-----------------------------------//
      //results = result;

      return;
    }     //End of dofull_process routine

745 float[] makeKernel(int radius) {
      int size = (int)radius*2-1;
      float[] kernel = new float[(int)sqr(size)];
      for (int i=0; i<size; i++){
        for (int j=0; j<size; j++) {
          kernel[i+size*j] = (float)Math.exp(-3.125*(sqr(i+1-radius)+sqr(j+1-radius)
              )/(sqr(radius)));
          if(debug==1) IJ.showMessage("i : " + i + "\nj : " + j + "\nKernel["+(i +
              size*j) +"] : " + kernel[i+size*j]);
        }
      }
      if (kernel.length==1) kernel[0] = 1f;
755   return kernel;
    }     //End of makeKernel routine

    double sqr(double x) {return x*x;}

    float sqr(float x) {return x*x;}

    void doConvolve(ImagePlus Iimp, ImagePlus Kimp, float[] dataO){

      ImageProcessor Iipi = Iimp.getProcessor();
765   ImageProcessor Kipi = Kimp.getProcessor();

      ImageProcessor Iip = Iipi.convertToFloat();
      ImageProcessor Kip = Kipi.convertToFloat();

      int Iw = Iip.getWidth();
      int Ih = Iip.getHeight();
      float[] dataIin = new float[Iw*Ih];
      dataIin = (float[])Iip.getPixels();
      if(debug==2) for (int i=0; i<dataIin.length; i++) IJ.write(IJ.d2s((double)
          dataIin[i]));
775
      int Kw = Kip.getWidth();
      int Kh = Kip.getHeight();
      float[] dataKin = new float[Kw*Kh];
      dataKin = (float[])Kip.getPixels();
      if(debug==2) for (int i=0; i<dataKin.length; i++) IJ.write(IJ.d2s((double)
          dataKin[i]));

      float scalePSF = 1;
      float sum = 0;
      for (int ind = 0; ind < Kh*Kw; ind++){
785     sum += dataKin[ind];
      }
      if(sum != 0)scalePSF /= sum;
      if(debug==2) IJ.write("Sum\t"+IJ.d2s((double)sum,4)+"\tScalePSF\t"+IJ.d2s((
          double)scalePSF,4));


      int IwE = expandedSize(Iw);
      int IhE = expandedSize(Ih);
      int KwE = expandedSize(Kw);
      int KhE = expandedSize(Kh);
795   if(debug==2) IJ.write("Iw\t"+Iw+"\tIwE\t"+IwE);
```

```
        if(debug==2) IJ.write("Ih\t"+Ih+"\tIhE\t"+IhE);
        if(debug==2) IJ.write("Kw\t"+Kw+"\tKwE\t"+KwE);
        if(debug==2) IJ.write("Kh\t"+Kh+"\tKhE\t"+KhE);
        //w and h will always be at least 4.
        int w = (int)Math.max(IwE,KwE);
        int h = (int)Math.max(IhE,KhE);
        if(debug==2) IJ.write("h\t"+h+"\tw\t"+w);

        IJ.showStatus("Creating expanded arrays " + IJ.d2s((double)i,0) + "/" + IJ.d2s
            ((double)totalruns,0));
805     float[] dataK = new float[w*h];
        copyDataMask(Kw,Kh,dataKin,w,h,dataK);
        //if(debug==2) for (int i=0; i<dataK.length; i++) IJ.write(IJ.d2s((double)
            dataK[i]));
        if(debug==2) {
          ImageProcessor tempip = new FloatProcessor(w,h);
          float[] temp = new float[w*h];
          for (int i=0; i<dataK.length; i++) temp[i] = dataK[i];
          tempip.setPixels(temp);
          ImagePlus tempimp = new ImagePlus("Kernel Expanded",tempip);
          tempimp.show();
815     }

        float[] dataI = new float[w*h];
        copyDataMask(Iw,Ih,dataIin,w,h,dataI);
        //if(debug==2) for (int i=0; i<dataI.length; i++) IJ.write(IJ.d2s((double)
            dataI[i]));
        if(debug==2) {
          ImageProcessor tempip = new FloatProcessor(w,h);
          float[] temp = new float[w*h];
          for (int i=0; i<dataI.length; i++) temp[i]=dataI[i];
          tempip.setPixels(temp);
825       ImagePlus tempimp = new ImagePlus("Image Expanded",tempip);
          tempimp.show();
        }

        IJ.showStatus("Swapping quadrants of the Kernel " + IJ.d2s((double)i,0) + "/"
            + IJ.d2s((double)totalruns,0));
        swapQuadrants(w,h,dataK);
        //if(debug==2) for (int i=0; i<dataK.length; i++) IJ.write(IJ.d2s((double)
            dataK[i]));
        if(debug==2) {
          ImageProcessor tempip = new FloatProcessor(w,h);
          float[] temp = new float[w*h];
835       for (int i=0; i<dataK.length; i++) temp[i]=dataK[i];
          tempip.setPixels(temp);
          ImagePlus tempimp = new ImagePlus("Kernel QsSwapped",tempip);
          tempimp.show();
        }

        //Add extra dimension to images to allow used of convolve3D plugin
        int d=1;
        float[][] dataK2 = new float[d][w*h];
        float[][] dataI2 = new float[d][w*h];
845     for (int ind = 0; ind < h*w; ind++){
          dataK2[0][ind] = dataK[ind];
          dataI2[0][ind] = dataI[ind];
        }
        if(debug==2) {
          ImageProcessor tempip = new FloatProcessor(w,h);
          float[] temp = new float[w*h];
          for (int i=0; i<dataK2[0].length; i++) temp[i]=dataK2[0][i];
```

```
        tempip.setPixels(temp);
        ImagePlus tempimp = new ImagePlus("Kernel 3D",tempip);
        tempimp.show();
      }
      if(debug==2) {
        ImageProcessor tempip = new FloatProcessor(w,h);
        float[] temp = new float[w*h];
        for (int i=0; i<dataI2[0].length; i++) temp[i]=dataI2[0][i];
        tempip.setPixels(temp);
        ImagePlus tempimp = new ImagePlus("Image 3D",tempip);
        tempimp.show();
      }

      float[][] result2 = new float[d][w*h];

      IJ.showStatus("Transforming Kernel " + IJ.d2s((double)i,0) + "/" + IJ.d2s((
          double)totalruns,0));
      FHT3D(dataK2,w,h,d,false);
      IJ.showStatus("Transforming Image " + IJ.d2s((double)i,0) + "/" + IJ.d2s((
          double)totalruns,0));
      FHT3D(dataI2,w,h,d,false);
      IJ.showStatus("Convolving in frequency domain " + IJ.d2s((double)i,0) + "/" +
          IJ.d2s((double)totalruns,0));
      convolveFD(w,h,d,dataK2,dataI2,result2);
      IJ.showStatus("Transforming result " + IJ.d2s((double)i,0) + "/" + IJ.d2s((
          double)totalruns,0));
      FHT3D(result2,w,h,d,true);
      if(debug==2) {
        ImageProcessor tempip = new FloatProcessor(w,h);
        float[] temp = new float[w*h];
        for (int i=0; i<result2[0].length; i++) temp[i]=result2[0][i];
        tempip.setPixels(temp);
        ImagePlus tempimp = new ImagePlus("Result2",tempip);
        tempimp.show();
      }

      //Remove unnecessary dimension from result
      float[] result = new float [w*h];
      for (int ind = 0; ind < h*w; ind++){
        result[ind] = result2[0][ind];
      }
      if(debug==2) {
        ImageProcessor tempip = new FloatProcessor(w,h);
        float[] temp = new float[w*h];
        for (int i=0; i<result.length; i++) temp[i]=result[i];
        tempip.setPixels(temp);
        ImagePlus tempimp = new ImagePlus("Result",tempip);
        tempimp.show();
      }

      //Apply scale factors
      if(scalePSF != 1){
        IJ.showStatus("Normalizing " + IJ.d2s((double)i,0) + "/" + IJ.d2s((double)
            totalruns,0));
        for (int ind = 0; ind < h*w; ind++){
          result[ind] *= scalePSF;
        }
      }
      if(debug==2) {
        ImageProcessor tempip = new FloatProcessor(w,h);
        float[] temp = new float[w*h];
        for (int i=0; i<result.length; i++) temp[i]=result[i];
```

```
        tempip.setPixels(temp);
        ImagePlus tempimp = new ImagePlus("Result Scaled",tempip);
        tempimp.show();
      }

915   //Crop the output to the size of Yin
      int jOff = (h - Ih + 1)/2;
      int iOff = (w - Iw + 1)/2;
      float norm = (float)Math.sqrt(w*h);
      for (int j = 0; j < Ih; j++){
        for (int i = 0; i < Iw; i++){
          dataO[i + Iw*j] = result[i + iOff + w*(j+jOff)];
        }
      }
      if(debug==2) {
925     ImageProcessor tempip = new FloatProcessor(Iw,Ih);
        float[] temp = new float[Iw*Ih];
        for (int i=0; i<dataO.length; i++) temp[i]=dataO[i];
        tempip.setPixels(temp);
        ImagePlus tempimp = new ImagePlus("Data Out",tempip);
        tempimp.show();
      }

      return;
    }     //End of doConvolve routine
935
    void copyDataMask(int w, int h,float[] data,int wE,int hE,float[] dataE){
      int jOff = (hE - h + 1)/2;
      int iOff = (wE - w + 1)/2;
      for (int j = 0; j < h; j++){
        for (int i = 0; i < w; i++){
          dataE[i+iOff + wE*(j+jOff)] = data[i + w*j];
        }
      }
    }     //End of copyDataMask routine
945
    //A version of mod that is periodic for postive and negative i
    int mod(int i, int n){
      return ((i % n) + n) % n;
    }     //End of mod routine

    int expandedSize(int maxN){
      //Expand this to a power of 2 that is at least 1.5* as large, to avoid wrap
          effects
      //Start with 4 to avoid apparent normalization problems with n = 2
      int iN=4;
955   if(maxN > 1){
        while(iN<1.5 * maxN) iN *= 2;
      }
      return iN;
    }     //End of expandedSize routine

    void swapQuadrants(int w,int h,float[] x){
      int k1P,k2P;
      float temp;
      int wHalf = w/2;
965   int hHalf = h/2;
      //Shift by half of the grid, less one pixel, in each direction
      for(int k2 = 0; k2 < hHalf; k2++){
        k2P = k2 + hHalf;
        for (int k1 = 0; k1 < w; k1++){
          temp = x[k1 + w*k2];
```

```
            x[k1 + w*k2] = x[k1 + w*k2P];
            x[k1 + w*k2P] = temp;
          }
        }
        for(int k1 = 0; k1 < wHalf; k1++){
          k1P = k1 + wHalf;
          for (int k2 = 0; k2 < h; k2++){
            temp = x[k1 + w*k2];
            x[k1 + w*k2] = x[k1P + w*k2];
            x[k1P + w*k2] = temp;
          }
        }
      }      //End of swapQuadrants routine

      void convolveFD(int w,int h,int d,float[][] h1,float[][] h2, float[][] result){
        int k1C,k2C,k3C;
        double h2e,h2o;
        for(int k3 = 0; k3 < d; k3++){
          k3C = (d - k3) % d;
          for (int k2 = 0; k2 < h; k2++){
            k2C = (h - k2) % h;
            for (int k1 = 0; k1 < w; k1++){
              k1C = (w - k1) % w;
              h2e = (h2[k3][k1 + w*k2] + h2[k3C][k1C + w*k2C])/2;
              h2o = (h2[k3][k1 + w*k2] - h2[k3C][k1C + w*k2C])/2;
              result[k3][k1 + w*k2] = (float)(h1[k3][k1 + w*k2]*h2e + h1[k3C][k1C + w*
                  k2C]*h2o);
            }
          }
        }
      }      //End of convolveFD routine

      boolean powerOf2Size(int w) {
        int i=2;
        while(i<w) i *= 2;
        return i==w;
      }      //End of powerOf2Size routine

      void FHT3D(float[][] data,int w, int h, int d, boolean inverse) {
        float[] sw = new float[w/4];
        float[] cw = new float[w/4];
        float[] sh = new float[h/4];
        float[] ch = new float[h/4];
        makeSinCosTables(w,sw,cw);
        makeSinCosTables(h,sh,ch);
        for (int i = 0; i < d; i++){
          rc2DFHT(data[i], w, h, sw, cw, sh, ch);
        }
        float[] u = new float[d];
        //if(IJ.getNumber("0 for fast, 1 for slow",0)==0){
        if(powerOf2Size(d)){
          float[] s = new float[d/4];
          float[] c = new float[d/4];
          makeSinCosTables(d,s,c);
          for(int k2 = 0; k2 < h; k2++){
            for(int k1 = 0; k1 < w; k1++){
              int ind = k1 + k2*w;
              for(int k3 = 0; k3 < d; k3++){
                u[k3] = data[k3][ind];
              }
              dfht3(u, 0, d, s, c);
              for(int k3 = 0; k3 < d; k3++){
```

```
            data[k3][ind] = u[k3];
          }
        }
      }
    }else{
      float[] cas = hartleyCoefs(d);
      float[] work = new float[d];
      for(int k2 = 0; k2 < h; k2++){
        for(int k1 = 0; k1 < w; k1++){
          int ind = k1 + k2*w;
          for(int k3 = 0; k3 < d; k3++){
            u[k3] = data[k3][ind];
          }
          slowHT(u,cas,d,work);
          for(int k3 = 0; k3 < d; k3++){
            data[k3][ind] = u[k3];
          }
        }
      }
    }
    //Convert to actual Hartley transform
    float A,B,C,D,E,F,G,H;
    int k1C,k2C,k3C;
    for(int k3 = 0; k3 <= d/2; k3++){
      k3C = (d - k3) % d;
      for(int k2 = 0; k2 <= h/2; k2++){
        k2C = (h - k2) % h;
        for (int k1 = 0; k1 <= w/2; k1++){
          k1C = (w - k1) % w;
          A = data[k3][k1 + w*k2C];
          B = data[k3][k1C + w*k2];
          C = data[k3C][k1 + w*k2];
          D = data[k3C][k1C + w*k2C];
          E = data[k3C][k1 + w*k2C];
          F = data[k3C][k1C + w*k2];
          G = data[k3][k1 + w*k2];
          H = data[k3][k1C + w*k2C];
          data[k3][k1 + w*k2] = (A+B+C-D)/2;
          data[k3C][k1 + w*k2] = (E+F+G-H)/2;
          data[k3][k1 + w*k2C] = (G+H+E-F)/2;
          data[k3C][k1 + w*k2C] = (C+D+A-B)/2;
          data[k3][k1C + w*k2] = (H+G+F-E)/2;
          data[k3C][k1C + w*k2] = (D+C+B-A)/2;
          data[k3][k1C + w*k2C] = (B+A+D-C)/2;
          data[k3C][k1C + w*k2C] = (F+E+H-G)/2;
        }
      }
    }
    if(inverse){
      //float norm = (float)Math.sqrt(d*h*w);
      float norm = d*h*w;
      for(int k3 = 0; k3 < d; k3++){
        for(int k2 = 0; k2 < h; k2++){
          for (int k1 = 0; k1 < w; k1++){
          data[k3][k1 + w*k2] /= norm;
          }
        }
      }
    }
  }     //End of FHT3D routine

  float[] hartleyCoefs(int max){
```

```
        float[] cas = new float[max*max];
        int ind = 0;
        for(int n = 0; n < max; n++){
          for (int k = 0; k < max; k++){
            double arg = (2*Math.PI*k*n)/max;
            cas[ind++] = (float)(Math.cos(arg) + Math.sin(arg));
          }
        }
        return cas;
      }     //End of hartleyCoefs routine

      void slowHT(float[] u, float[] cas, int max, float[] work){
        int ind = 0;
        for(int k = 0; k < max; k++){
          float sum = 0;
          for(int n = 0; n < max; n++){
            sum += u[n]*cas[ind++];
          }
          work[k] = sum;
        }
        for (int k = 0; k < max; k++){
          u[k] = work[k];
        }
      }     //End of slowHT routine

      void makeSinCosTables(int maxN, float[] s, float[] c) {
        int n = maxN/4;
        double theta = 0.0;
        double dTheta = 2.0 * Math.PI/maxN;
        for (int i=0; i<n; i++) {
          c[i] = (float)Math.cos(theta);
          s[i] = (float)Math.sin(theta);
          theta += dTheta;
        }
      }     //End of makeSinCosTables routine

      // Row-column Fast Hartley Transform
      void rc2DFHT(float[] x, int w, int h, float[] sw, float[] cw, float[] sh, float
          [] ch) {
        for (int row=0; row<h; row++)
          dfht3(x, row*w, w, sw, cw);
        float[] temp = new float[h];
        for(int col = 0; col < w; col++){
          for (int row = 0; row < h; row++){
            temp[row] = x[col + w*row];
          }
          dfht3(temp, 0, h, sh, ch);
          for (int row = 0; row < h; row++){
            x[col + w*row] = temp[row];
          }
        }
      }     //End of rc2DFHT routine

      // An optimized real FHT
      void dfht3(float[] x, int base, int maxN, float[] s, float[] c) {
        int i, stage, gpNum, gpIndex, gpSize, numGps, Nlog2;
        int bfNum, numBfs;
        int Ad0, Ad1, Ad2, Ad3, Ad4, CSAd;
        float rt1, rt2, rt3, rt4;

        Nlog2 = log2(maxN);
        BitRevRArr(x, base, Nlog2, maxN); //bitReverse the input array
```

```
1155      gpSize = 2;      //first & second stages - do radix 4 butterflies once thru
          numGps = maxN / 4;
          for (gpNum=0; gpNum<numGps; gpNum++)  {
            Ad1 = gpNum * 4;
            Ad2 = Ad1 + 1;
            Ad3 = Ad1 + gpSize;
            Ad4 = Ad2 + gpSize;
            rt1 = x[base+Ad1] + x[base+Ad2];    // a + b
            rt2 = x[base+Ad1] - x[base+Ad2];    // a - b
            rt3 = x[base+Ad3] + x[base+Ad4];    // c + d
1165        rt4 = x[base+Ad3] - x[base+Ad4];    // c - d
            x[base+Ad1] = rt1 + rt3;       // a + b + (c + d)
            x[base+Ad2] = rt2 + rt4;       // a - b + (c - d)
            x[base+Ad3] = rt1 - rt3;       // a + b - (c + d)
            x[base+Ad4] = rt2 - rt4;       // a - b - (c - d)
          }
          if (Nlog2 > 2) {
             // third + stages computed here
            gpSize = 4;
            numBfs = 2;
1175        numGps = numGps / 2;
            //IJ.write("FFT: dfht3 "+Nlog2+" "+numGps+" "+numBfs);
            for (stage=2; stage<Nlog2; stage++) {
              for (gpNum=0; gpNum<numGps; gpNum++) {
                Ad0 = gpNum * gpSize * 2;
                Ad1 = Ad0;      // 1st butterfly is different from others - no mults
                    needed
                Ad2 = Ad1 + gpSize;
                Ad3 = Ad1 + gpSize / 2;
                Ad4 = Ad3 + gpSize;
                rt1 = x[base+Ad1];
1185            x[base+Ad1] = x[base+Ad1] + x[base+Ad2];
                x[base+Ad2] = rt1 - x[base+Ad2];
                rt1 = x[base+Ad3];
                x[base+Ad3] = x[base+Ad3] + x[base+Ad4];
                x[base+Ad4] = rt1 - x[base+Ad4];
                for (bfNum=1; bfNum<numBfs; bfNum++) {
                // subsequent BF's dealt with together
                  Ad1 = bfNum + Ad0;
                  Ad2 = Ad1 + gpSize;
                  Ad3 = gpSize - bfNum + Ad0;
1195              Ad4 = Ad3 + gpSize;

                  CSAd = bfNum * numGps;
                  rt1 = x[base+Ad2] * c[CSAd] + x[base+Ad4] * s[CSAd];
                  rt2 = x[base+Ad4] * c[CSAd] - x[base+Ad2] * s[CSAd];

                  x[base+Ad2] = x[base+Ad1] - rt1;
                  x[base+Ad1] = x[base+Ad1] + rt1;
                  x[base+Ad4] = x[base+Ad3] + rt2;
                  x[base+Ad3] = x[base+Ad3] - rt2;
1205
                } // end bfNum loop
              } // end gpNum loop
              gpSize *= 2;
              numBfs *= 2;
              numGps = numGps / 2;
            } // end for all stages
          } // end if Nlog2 > 2
        }     //End of dfht3 routine

1215    int log2(int x) {
```

```
    int count = 15;
    while (!btst(x, count))
      count--;
    return count;
  }     //End of log2 routine

  boolean btst(int  x, int bit) {
    //int mask = 1;
    return ((x & (1<<bit)) != 0);
  }     //End of btst routine

  void BitRevRArr(float[] x, int base, int bitlen, int maxN) {
    int     l;
    float[] tempArr = new float[maxN];
    for (int i=0; i<maxN; i++)  {
      l = BitRevX (i, bitlen);  //i=1, l=32767, bitlen=15
      tempArr[i] = x[base+l];
    }
    for (int i=0; i<maxN; i++)
      x[base+i] = tempArr[i];
  }     //End of BitRevRArr routine

  int BitRevX(int  x, int bitlen) {
    int  temp = 0;
    for (int i=0; i<=bitlen; i++)
      if ((x & (1<<i)) !=0)
        temp  |= (1<<(bitlen-i-1));
    return temp & 0x0000ffff;
  }     //End of BitRevX routine

  int bset(int x, int bit) {
    x |= (1<<bit);
    return x;
  }     //End of bset routine

  ImageStack dosurfaceplot(ImageStack is) {

    ImageStack stack;
    ImageProcessor ip, plot;
    int i, n;

    n = is.getSize();
    ip = is.getProcessor(1);
    IJ.showStatus("Surface Plot: Slice 1/" + n);
    plot = makeSurfacePlot(ip);
    stack = new ImageStack(plot.getWidth(), plot.getHeight());
    stack.addSlice("1", plot);

    for(i = 2; i <= n; i++) {
      ip = is.getProcessor(i);
      IJ.showStatus("Surface Plot: Slice " + i + "/" + n);
      plot = makeSurfacePlot(ip);
      stack.addSlice(IJ.d2s(i,0), plot);
    }

    return stack;
  }

  ImageProcessor makeSurfacePlot(ImageProcessor ip) {
    ip = ip.duplicate();
    Rectangle roi = img.getProcessor().getRoi();
    ip.setRoi(roi);
```

```
        if (!(ip instanceof ByteProcessor)) {
          ip.setMinAndMax(img.getProcessor().getMin(), img.getProcessor().getMax());
          ip = ip.convertToByte(true);
          ip.setRoi(roi);
        }
        double angle = (angleInDegrees/360.0)*2.0*Math.PI;
        int polygons = (int)(plotWidth*(polygonMultiplier/100.0)/4);
1285    if (oneToOne)
          polygons = roi.height;
        double xinc = 0.8*plotWidth*Math.sin(angle)/polygons;
        double yinc = 0.8*plotWidth*Math.cos(angle)/polygons;
        IJ.showProgress(0.01);
        ip.setInterpolate(!oneToOne);
        ip = ip.resize(plotWidth, polygons);
        int width = ip.getWidth();
        int height = ip.getHeight();
        double min = ip.getMin();
1295    double max = ip.getMax();

        if(invertedLut) ip.invert();
        if(whiteBackground) ip.invert();
        if (smooth) ip.smooth();

        x = new int[width+2];
        y = new int[width+2];
        double xstart = 10.0;
        if (xinc<0.0)
1305      xstart += Math.abs(xinc)*polygons;
        ByteProcessor ipProfile =new ByteProcessor(width, (int)(256+width*yinc2));
        ipProfile.setValue(255);
        ipProfile.fill();
        double ystart =  yinc2*width;
        int ybase = (int)(ystart+0.5);
        int windowWidth =(int)(plotWidth+polygons*Math.abs(xinc) + 20.0);
        int windowHeight = (int)(ipProfile.getHeight()+polygons*yinc + 10.0);

        if(showAxis){
1315      xstart += 50+20;
          ystart += 10;
          windowWidth += 60+20;
          windowHeight += 20;
          p1x = xstart;
          p1y = ystart+255;
          p2x = xstart+xinc*height;;
          p2y = p1y+yinc*height;
          p3x =  p2x+width-1;
          p3y = p2y- yinc2*width;
1325    }

        if(showGrayscale) {
          int v;
          int[] column = new int[255];
          for(int row=0; row<255; row++) {
            if(whiteBackground)
              v = row;
            else
              v = 255-row;
1335        column[row] = v;
          }
          int base = ipProfile.getHeight()-255;
          for(int col=0; col<width; col++) {
            ipProfile.putColumn(col, base-(int)(yinc2*col+0.5), column, 255);
```

```
        }
      } else {
        ipProfile.setValue(254);
        ipProfile.fill();
      }

      ipProfile.snapshot();

      ImageProcessor ip2 = new ByteProcessor(windowWidth, windowHeight);
      if(showGrayscale) {
        ip2.setColorModel(ip.getColorModel());
        if(invertedLut)
          ip2.invertLut();
        fixLut(ip2);
      }
      if(!blackFill)
        ip2.setValue(255);
      else
        ip2.setValue(0);
      ip2.fill();

      for (int row=0; row<height; row++) {
        double[] profile = ip.getLine(0, row, width-1, row);
        clearAboveProfile(ipProfile, profile, width, yinc2);
        int ixstart = (int)(xstart+0.5);
        int iystart = (int)(ystart+0.5);

        ip2.copyBits(ipProfile, ixstart, iystart-ybase, Blitter.COPY_TRANSPARENT);
        ipProfile.reset();

        if (showWireframe) {
          ip2.setValue(0);
          double ydelta = 0.0;
          ip2.moveTo(ixstart, (int)(ystart+255.5 - profile[0]) );
          for(int i=1; i<width; i++) {
            ydelta += yinc2;
            ip2.lineTo( ixstart+i,  (int) (ystart+255.5-(profile[i]+ydelta)));
          }
          ip2.drawLine(ixstart, iystart+255, ixstart + width-1, (int)( ystart+255.5-
              ydelta) );
          ip2.drawLine( ixstart, iystart+255-(int) (profile[0]+0.5), ixstart,
              iystart+255 );
          ip2.drawLine( ixstart+width-1, (int) ( ystart+255.5-ydelta), ixstart+
              width-1, (int) (ystart+255.5-(profile[width-1]+ydelta)) );
        }

        xstart += xinc;
        ystart += yinc;
        if ((row%10)==0) IJ.showProgress((double)row/height);
      }

      IJ.showProgress(1.0);

      if(invertedLut) {
        ip.invert();
        ip.invertLut();
      }
      if(whiteBackground)
        ip.invert();

      if (showAxis) {
        if (!lut.isGrayscale() && showGrayscale)
```

```
        ip2 = ip2.convertToRGB();
      drawAndLabelAxis(ip, ip2, roi);
    }

    if (img.getStackSize()==1)
      ip2 = trimPlot(ip2, ybase);

    return ip2;
  }

  void drawAndLabelAxis(ImageProcessor ip, ImageProcessor ip2, Rectangle roi) {
    if(!blackFill)
      ip2.setColor(Color.black);
    else
      ip2.setColor(Color.white);
    ip2.setAntialiasedText(true);
    String s;
    int w, h;
    Calibration cal = img.getCalibration();

    //z-axis & label
    s = cal.getValueUnit();
    if (s.equals("Gray Value"))
      s = "";
    w =  ip2.getFontMetrics().stringWidth(s);
    drawAxis(ip2, (int) p1x, (int) p1y-255, (int) p1x, (int) p1y , s, 10, -1, 0,
        1);
    double min, max;
    if (img.getBitDepth()==8) {
      min = 0;
      max = 255;
    } else {
      min = img.getProcessor().getMin();
      max = img.getProcessor().getMax();
    }
    //IJ.log("");
    //IJ.log(min+" "+max+" "+cal.getCValue((int)min)+" "+cal.getCValue((int)max
        ));
    //ip2.putPixelValue(0,0,0);
    //boolean zeroIsBlack
    //IJ.log(ip2.getPixelValue(+" "+max+" "+cal.getCValue((int)min)+" "+cal.
        getCValue((int)max));
    if (cal.calibrated()) {
      min = cal.getCValue((int)min);
      max = cal.getCValue((int)max);
    }
    //if (invertedPixelValues)
    //  {double t=max; max=min; min=t;}
    ip2.setAntialiasedText(true);
    s = String.valueOf( (double) Math.round(max*10)/10);
    w =  ip.getFontMetrics().stringWidth(s);
    h =  ip.getFontMetrics().getHeight();
    ip2.drawString(s, (int) p1x-15-w, (int) p1y-255 +h/2);  //ybase+5+h+(int (
        yinc2/xinc *10));
    s = String.valueOf( (double) Math.round(min*10)/10);
    w =  ip2.getFontMetrics().stringWidth(s);
    ip2.drawString(s, (int) p1x-15-w, (int) p1y +h/2);

    //x-axis
    s = (double) Math.round(roi.height*cal.pixelHeight*10)/10+" "+cal.getUnits();
    w =  ip2.getFontMetrics().stringWidth(s);
    drawAxis(ip2, (int) p1x, (int) p1y, (int) p2x, (int) p2y, s, 10, -1, 1, 1);
```

```
    //y-axis
    s =  (double) Math.round(roi.width*cal.pixelWidth*10)/10+" "+cal.getUnits();
    w =  ip2.getFontMetrics().stringWidth(s);
    //drawAxis(ip2, (int) p2x, (int) p2y, (int) p3x, (int) p3y, s, 10, 1 , 1, -1);
    drawAxis(ip2, (int) p2x, (int) p2y, (int) p3x, (int) p3y, s, 10, 1, -1, 1);

  }

  void drawAxis(ImageProcessor ip, int x1, int y1, int x2, int y2, String label,
      int offset, int offsetXDirection, int offsetYDirection, int labelSide){
    if(blackFill)
      ip.setColor(Color.white);
    else
      ip.setColor(Color.black);

    double m = -(double) (y2-y1)/(double) (x2-x1);

    if(m==0)
      m=.0001;
    double mTangent = -1/m;
    double theta = Math.atan(mTangent);

    int dy = -offsetXDirection * (int) ( 7*Math.sin(theta) );
    int dx = -offsetXDirection * (int) ( 7*Math.cos(theta) );

    x1 += offsetXDirection * (int) ( offset*Math.cos(theta) );
    x2 += offsetXDirection * (int) ( offset*Math.cos(theta) );

    y1 += offsetYDirection * (int) ( offset*Math.sin(theta) );
    y2 += offsetYDirection * (int) ( offset*Math.sin(theta) );

    ip.drawLine(x1, y1, x2, y2);

    ip.drawLine(x1, y1, x1+dx, y1-dy);
    ip.drawLine(x2, y2, x2+dx, y2-dy);
    ImageProcessor ipText = drawString( ip, label, (int) (Math.atan(m)/2/Math.PI
        *360) );
    if(blackFill)
      ipText.invert();

    Blitter b;
    if (ip instanceof ByteProcessor)
      b = new ByteBlitter((ByteProcessor) ip);
    else
      b = new ColorBlitter((ColorProcessor) ip);
    Color c = blackFill?Color.black:Color.white;
    b.setTransparentColor(c);
    int xloc = (x1+x2)/2-ipText.getWidth()/2  + offsetXDirection*labelSide*(int)
        (15*Math.cos(theta));
    int yloc = (y1+y2)/2-ipText.getHeight()/2 + offsetYDirection*labelSide*(int)
        (15*Math.sin(theta));
    b.copyBits(ipText, xloc, yloc, Blitter.COPY_TRANSPARENT);

    return;
  }

  ImageProcessor drawString(ImageProcessor ip, String s, int a){
    int w =  ip.getFontMetrics().stringWidth(s);
    int h =  ip.getFontMetrics().getHeight();
    int ipW, ipH;
```

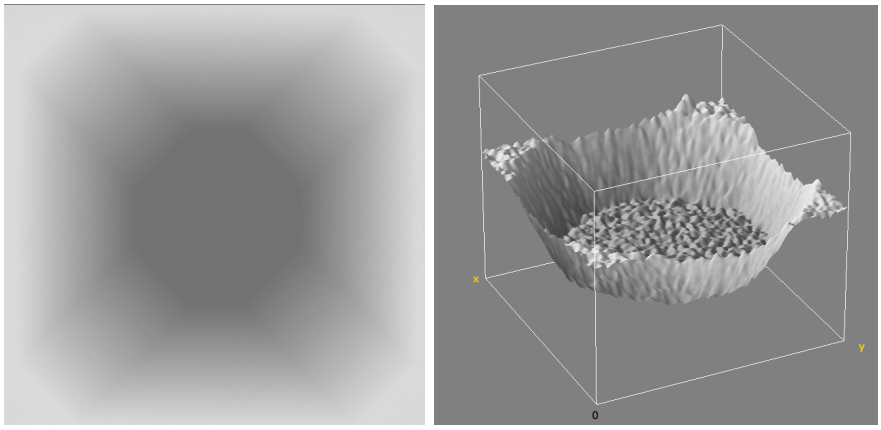
```
1515      double r = Math.sqrt( (w/2)*(w/2) + (h/2)*(h/2) );
          double aR = (a/360.0)*2.0*Math.PI;
          double aBaseR = Math.acos( (w/2)/r );

          ipW = (int) Math.abs(r*Math.cos(aBaseR+aR));
          ipH = (int) Math.abs(r*Math.sin(aBaseR+aR));

          if((int) Math.abs(r*Math.cos(-aBaseR+aR))>ipW)
            ipW = (int) Math.abs(r*Math.cos(-aBaseR+aR));
          if((int) Math.abs(r*Math.sin(-aBaseR+aR))>ipH)
1525        ipH = (int) Math.abs(r*Math.sin(-aBaseR+aR));

          ipW *= 2;
          ipH *= 2;

          int tW = w;
          if(ipW>w)
            tW = ipW;
          ImageProcessor ipText = new ByteProcessor(tW, ipH);
          ipText.setColor(Color.white);
1535      ipText.fill();
          ipText.setColor(Color.black);
          ipText.setAntialiasedText(true);
          ipText.drawString(s, tW/2-w/2, ipH/2+h/2);
          ipText.setInterpolate(true);
          ipText.rotate(-a);
          ipText.setRoi(tW/2-ipW/2, 0, ipW, ipH);
          ipText = ipText.crop();

          //new ImagePlus("test", ipText).show();
1545      //ip.copyBits(ipText, x, y, Blitter.COPY_TRANSPARENT);

          return ipText;
        }

        void clearAboveProfile(ImageProcessor ipProfile, double[] profile, int width,
            double yinc2) {
          byte[] pixels = (byte[])ipProfile.getPixels();
          double ydelta = 0.0;
          int height = ipProfile.getHeight();
          for(int x=0; x<width; x++) {
1555        ydelta += yinc2;
            int top = height - (int)(profile[x]+ydelta);
            for (int y=0,index=x; y<top; y++, index+=width)
              pixels[index] = (byte)255;
          }
        }

        ImageProcessor trimPlot(ImageProcessor plot, int maxTrim) {
          int background = plot.getPixel(0, 0);
          int width = plot.getWidth();
1565      int height = plot.getHeight();
          int trim = maxTrim-5;
      a: for (int y=0; y<(maxTrim-5); y++)
          for (int x=0; x<width; x++)
            if (plot.getPixel(x,y)!=background)
            {trim = y-5; break a;}
            if (trim>10) {
              plot.setRoi(0, trim, width, height-trim);
              plot = plot.crop();
            }
1575        return plot;
```

```
    }


    void fixLut(ImageProcessor ip) {
      if(!lut.isGrayscale() && lut.getMapSize() == 256){

        for(int y=0;y<ip.getHeight();y++){
          for(int x=0;x<ip.getWidth();x++){
            if(ip.getPixelValue(x, y)==0){
              ip.putPixelValue(x, y, 1);
            }else if(ip.getPixelValue(x, y)==255){
              ip.putPixelValue(x, y, 254);
            }

          }
        }

        byte[] rLUT = lut.getReds();       //new byte[256];
        byte[] gLUT = lut.getGreens();       //new byte[256];
        byte[] bLUT = lut.getBlues();      //new byte[256];

        rLUT[0] = (byte)0;
        gLUT[0] = (byte)0;
        bLUT[0] = (byte)0;
        rLUT[255] = (byte)255;
        gLUT[255] = (byte)255;
        bLUT[255] = (byte)255;

        ip.setColorModel(new IndexColorModel(8, 256, rLUT, gLUT, bLUT));

      }
    }

}
```

B Aligning to Topographical Markers

In micro-machined wafers it is common to use anisotropic etches to form three-dimensional structures on the wafer surface. It is desirable to be able to create alignment markers concurrently with other structures to reduce the cost and complexity of the fabrication process however the anisotropic etch process by its very nature does not produce sharp steps in the substrate. Instead the etch process produces sloped edges from the fast etching planes. These planes combine to produce inverted pyramids when the four planes terminate against each other. If the etch is stopped before the fast etching planes have terminated then truncated pyramids are formed. Interestingly the intensity of the backscattered electron (BSE) image of these markers is closely related to the topography of the silicon. While this does provide a usable amount of contrast in the BSE images the line scan techniques previously described in section 2.2.2 are inadequate to find the edges of these markers.

An example of a backscattered electron image of such a marker is shown in figure B.1(a) from this and the surface plot of figure B.1(b) it is very clear to see the link between the marker topography and the detected intensity in the BSE image. There are significant properties of the etch process which can be exploited to aid in the locating of the centre of the marker within the image. The angles of the sloped edges around the marker are very well defined by the crystal structure of the substrate furthermore the etch process naturally forms a symmetrical structure. Knowing that the underlying silicon structure has this symmetry and that the BSE signal reflects this geometry the



(a) Backscattered electron image of etched marker.

(b) Surface plot of marker image.

Figure B.1: Anisotropically etched topographical marker BSE image and the surface plot of that image showing the relation to the topography of the marker.

centre of the structure can be deduced from an acquired BSE image.

Figure B.2 shows the marker image with one grey level highlighted. By averaging the x and the y positions of all the highlighted pixels an estimate of the markers centre is gained. Repeating this over all the levels in the sloped edges of the marker and finally averaging all these results gives a precise measure of the centre of the marker. The accuracy of this method depends on the number of pixels across each edge and in the number of levels that span the sloped edge, but an alignment accuracy of 50 nm can be achieved. Although this is not superb, it is a vast improvement over the conventional line scanning methods which are often incapable of finding such markers.

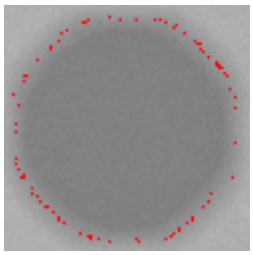


Figure B.2: Topographical marker with pixels from one grey level on the sloped edge of the marker highlighted.

C Published Work

- K.E. Docherty, S. Thoms, P. Dobson, J.M.R. Weaver, "Improvements to the alignment process in a commercial vector scan electron beam lithography tool" *Microelectronic Engineering*, vol. 85, pp. 761–763, 2008.
- K.E. Docherty, K.A. Lister, J. Romijn, J.M.R. Weaver, "High robustness of correlation-based alignment with Penrose patterns to marker damage in electron beam lithography", *Microelectronic Engineering*, vol. 86, pp. 532–534, 2009.

Available online at www.sciencedirect.com
ScienceDirect
Microelectronic Engineering 85 (2008) 761–763
MICROELECTRONIC ENGINEERING
www.elsevier.com/locate/mee

Improvements to the alignment process in a commercial vector scan electron beam lithography tool

K.E. Docherty *, S. Thoms, P. Dobson, J.M.R. Weaver

Nanoelectronics Research Centre, Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8LT, UK

Received 5 October 2007; received in revised form 16 December 2007; accepted 27 January 2008
Available online 2 February 2008

Abstract

This paper examines the desirable properties of marker patterns for use in correlation-based alignment systems and demonstrates alignment accuracies of better than 1 nm. A framework for evaluating different classes of marker patterns has been developed and a figure of merit for marker patterns used in correlation-based alignment has been defined. We show that Penrose tilings have many desirable properties for correlation-based alignment. An alignment system based on correlation and using marker patterns derived from Penrose tilings has been developed and implemented on a commercial Vistec VB6 UHR EWF electron beam lithography tool. A new method of measuring alignment at the sub-nm level using overlaid gratings and a Fourier Transform based analysis scheme is introduced.

Keywords: Electron beam lithography; Alignment; Correlation; Penrose tilings

1. Introduction

The accuracy of current methods of alignment has limited the use of lithographic techniques in the fabrication of some of the smallest devices, notably nanogaps [1]. Previous work has shown that alignment algorithms based on the process of correlation significantly improve the attainable accuracy [2] and that sub-pixel accuracy is achievable [3]. Although previous work [4,5] has shown that the marker pattern used is an important factor affecting the accuracy of correlation-based alignment the optimum design has not been investigated.

2. Marker patterns

For correlation-based alignment, the important feature of a marker pattern is a sharply peaked autocorrelation, the correlation of a function with itself. The sharpness of the autocorrelation peak represents its sensitivity to small positional shifts and hence the accuracy.

Several figures of merit have been defined for peak sharpness of autocorrelations [6], but the most applicable is the peak-to-correlation energy ratio PCE. This quantifies the sharpness as the ratio of the energy contained in the central pixel of the autocorrelation to the total energy in the correlation plane.

The best alignment would result from an autocorrelation that could be approximated by a 2D δ-function, i.e. PCE = 1. Using the Wiener–Khintchine theorem, that states that the autocorrelation of a function and the function's power spectral density are a Fourier transform pair [7], implies that good patterns should have a uniform sampling of frequency space. Thus the ideal shape is a perfectly aperiodic pattern, with a good sampling of spatial frequencies.

One good class of patterns are Penrose tilings [8] which are tilings based on only two fundamental tiles that can be used to perfectly cover an infinite plane aperiodically. The autocorrelations of Penrose patterns have subsidiary peaks around the central maxima but, since the Penrose pattern will be used to pull-in from an already good alignment, this limitation is traded for their superior peak sharpness.

* Corresponding author. Tel.: +44 141 330 8631.
E-mail address: kevin.docherty@elec.gla.ac.uk (K.E. Docherty).

0167-9317/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.mee.2008.01.081

Penrose patterns also exhibit a lack of coherence with a rectangular pixel array. This is advantageous since no matter where the pattern is sampled, small shifts in sampling position always produce significant changes in the sampled result. Whereas, patterns that have some coherence with a rectangular sampling grid, or some 'grid degeneracy' have areas where small shifts in sampling position produces no, or insignificant, changes in the result.

The grid degeneracy effects are most evident when undersampling the pattern. The tolerance of Penrose patterns to undersampling along with the linear nature of the correlation process allows interrogation area to be traded with beam current to maintain a good signal-to-noise ratio. For example, by exposing a larger area with a lower beam current we can collect enough information to locate the marker but not expose the surrounding resist. This yields a reduction in errors during multi-stage alignments.

Penrose patterns also exhibit approximately 50% fill, and are composed of many similarly sized primitive elements, maximising high spatial frequency components while maintaining compatibility with fabrication limits, making them an ideal choice for markers.

3. Implementation

To implement the method of alignment described, Penrose marker patterns have been fabricated from Ti/Au by lift-off on silicon wafers. Software has been developed and incorporated into the operation of a Vistec VB6 UHR EWF electron beam lithography tool to perform marker searches. The software controls the machine to collect a backscattered electron image of the marker and performs the correlation of this image with a previously collected reference image. The peak of the correlation is then least-squares fitted to a 2D Gaussian function using the Levenberg–Marquadt algorithm [9] to allow the marker offset to be calculated with sub-pixel precision.

Measurements were performed to compare the accuracy of the correlation method with the conventional mark locate method. A pair of Penrose markers and a pair of standard markers were used and on each iteration a measure of the separation between each pair of markers was obtained by doing marker searches. This allowed systematic errors common to both pairs to be eliminated, in particular it allowed the stage drift to be removed from the data. This leaves a measure of the inaccuracies of locating the marker, including the inaccuracies of the algorithm used. The graph in Fig. 1 show the results of this comparison for the x-direction. Similar results (not shown) were obtained for the y-direction. The σ values quoted on the graph gives the standard deviation of each of the curves and since the curves only vary with errors in the algorithm this gives a measure of the attainable accuracy of the correlation and conventional mark locate routines. In comparison with the mark locate results the correlation algorithm produces around ten times better accuracy, having a combined σ value for both x- and y-directions of 0.95 nm compared with 11 nm.

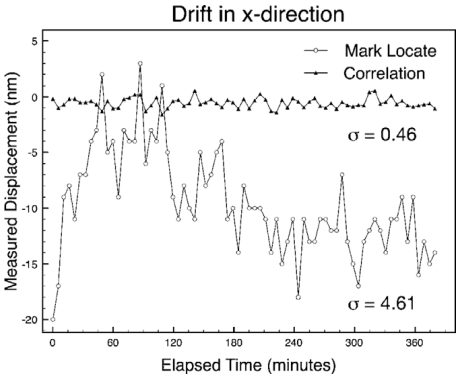


Fig. 1. Comparison between the drift variations of the mark locate and correlation-based methods in the x-direction, showing an estimate of the attainable accuracy with both methods and the 10-fold increase in accuracy with the correlation-based method.

3.1. Metrology

The need to be able to measure alignment accuracy to this level of precision posses significant problems. We propose the use of overlaid gratings, where the first half-grating of period $\frac{2}{f}$ is written, then the second half-grating also of period $\frac{2}{f}$ is written offset by $\frac{1}{f}$ from the first, to give an overall grating of period $\frac{1}{f}$. When the two halves of the grating are perfectly aligned a Fourier analysis of the grating shows that only the fundamental frequency, f, is present in the power spectrum. However, any misalignment between the first and the second half of the grating results in a component at the half-frequency, $\frac{f}{2}$ and, as shown in

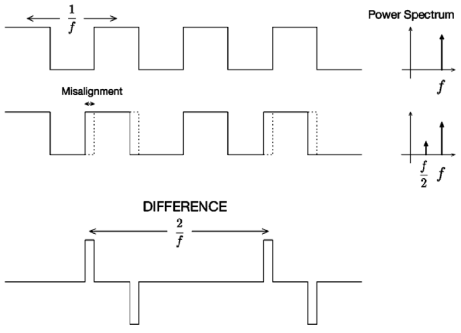


Fig. 2. Fourier analysis of an overlaid grating showing that the amplitude of the $\frac{f}{2}$ component in the power spectrum is directly proportional to the misalignment between the two halves of the grating.

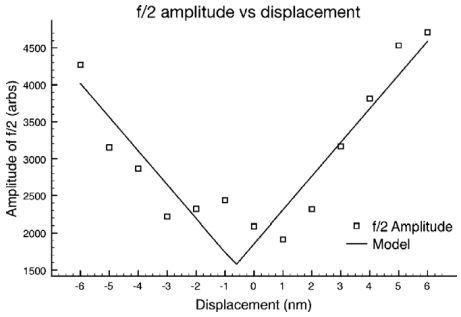


Fig. 3. Graph showing the amplitude of the $\frac{f}{2}$ component of the power spectrum plotted against offset, with model fitted demonstrating alignment measured at −0.63 nm.

Fig. 2, the amplitude of the $\frac{f}{2}$ component is directly proportional to the misalignment.

Writing gratings with a range of deliberate misalignments, analysing each grating, and plotting the magnitude of the $\frac{f}{2}$ peak against the deliberate offset gives a 'V'-shaped curve that can be used to measure the misalignment.

Gratings (100 nm period) have been written in HSQ in this fashion, imaged in an Hitachi S4700 SEM, then analysed. The resulting plot of half-frequency amplitude against induced offset is shown in Fig. 3. A least-squares fitting has been performed on this data to a model of two straight lines with common slope and from their intercept the misalignment is measured to be 0.63 nm.

4. Conclusions

Correlation-based alignment provides flexibility over conventional alignment methods allowing various marker patterns to be used. Penrose patterns demonstrate many desirable properties for marker patterns in correlation-based alignment. Their compatibility with fabrication techniques, significant high spatial frequency content and tolerance of rectangular sampling make them ideal patterns for alignment markers. Overlaid gratings with Fourier transform analysis provide a means to measure alignment at the sub-nm level. Alignment algorithms based on correlation and using Penrose patterns have been used to perform alignments with a measured accuracy of 0.63 nm.

References

[1] P. Steinmann, J.M.R. Weaver, Appl. Phys. Lett. 86 (6) (2005) 063104.
[2] V. Boegli, D.P. Kern, J. Vac. Sci. Technol. B 8 (6) (1990) 1994–2001.
[3] E.H. Anderson, D. Ha, J.A. Liddle, Microelectron. Eng. 73–74 (2004) 74–79.
[4] A.M. Bruckstein, L. O'Gorman, A. Orlitsky, IEEE Trans. Inform. Theory 44 (7) (1998) 3156–3162.
[5] Y.H. Chen, W.H. Huang, X.M. Dang, Rev. Sci. Instrum. 74 (7) (2003) 3549–3553.
[6] B.V.K.V. Kumar, L. Hassebrook, Appl. Opt. 29 (20) (1990) 2997–3006.
[7] R.N. Bracewell, The Fourier Transform and its Applications, second ed., McGraw-Hill, London, 1986.
[8] R. Penrose, The Emperor's New Mind: Concerning Computers, Minds and the Laws of Physics, Oxford University Press, 1989.
[9] M. Lourakis, levmar:levenberg-marquardt nonlinear least squares algorithms in C/C++, July 2004.

Microelectronic Engineering 86 (2009) 532–534

Contents lists available at ScienceDirect

Microelectronic Engineering

journal homepage: www.elsevier.com/locate/mee

High robustness of correlation-based alignment with Penrose patterns to marker damage in electron beam lithography

K.E. Docherty [a,*], K.A. Lister [b], J. Romijn [c], J.M.R. Weaver [a]

[a] James Watt Nanofabrication Centre, Department of Electronics and Electrical Engineering, University of Glasgow, Oakfield Avenue, Glasgow G12 8LT, UK
[b] Center of MicroNanoTechnology, École Polytechnique, Fédéral du Lausanne, CH-1015 Lausanne, Switzerland
[c] Vistec Lithography BV, De Dintel 27a, 5684 PS Best, The Netherlands

ARTICLE INFO

Article history:
Received 29 September 2008
Received in revised form 20 November 2008
Accepted 20 November 2008
Available online 3 December 2008

Keywords:
Electron beam lithography
Alignment
Correlation
Penrose patterns
Marker damage

ABSTRACT

Correlation-based alignment is an alternative alignment method for electron beam lithography. Using complex marker patterns, such as Penrose patterns, which contain more positional information, greater alignment accuracy can be achieved. Correlation-based alignment with Penrose patterns is less susceptible to marker edge defects, such as rat bites, roughness and flagging, since many more edges contribute to determining the marker position. There are however other defects associated with fabricating markers and this paper investigates how defects that result in parts of the pattern being omitted or obscured affect the correlation process when using Penrose pattern markers. We show that in both cases severely damaged markers can be used successfully and demonstrate fabricated structures with sub-5 nm alignment using markers with up to 80% of the marker pattern missing.

1. Introduction

Correlation is an image processing technique which has previously been used as an alignment method for electron beam lithography (ebeam) [1]. It has been shown to provide substantial improvements to the attainable accuracy with the scale of this gain governed by the choice of marker pattern [2–4]. Previous work [5] showed that Penrose patterns have numerous features that make them ideal patterns for markers and demonstrated that correlation-based alignment with Penrose patterns could be used to fabricate structures with sub-nm alignment accuracy. Software to perform correlation-based marker searches has been written and fully incorporated into the control software for both a Vistec VB6 UHR EWF and a Vistec EBPG5000+ ES HR which were used throughout this work.

1.1. Penrose patterns

Penrose patterns are infinite aperiodic tilings of the 2D plane formed from two fundamental shapes. They have a five fold rotational symmetry and crucially any region of the pattern is translationally unique within a given area. They are formed from many similarly sized elements and exhibit ~50% fill which maximises the number of edges within the pattern. Finally, since the fundamental elements are triangular and rhombic the pattern is incoherent with a rectangular sampling grid such that very small changes in sampling position always provide significant changes in the detected pattern and hence sensitivity to misalignment is relatively uniform. A potential disadvantage to the use of Penrose patterns as alignment marks is the complexity of fabrication. The marker ideally consist of many elements fabricated near to the resolution limit of the process used. It is not immediately clear how errors in the reproduction of the markers would affect the alignment accuracy. Traditional ebeam alignment using uniformly filled polygonal markers is highly susceptible to edge defects such as rat bites, roughness and flagging. In correlation-based alignment of Penrose pattern markers, there is expected to be less susceptibility to these edge effects since many more edges contribute to determining the marker position. There are other defects associated with fabricating marker patterns and this paper investigates how two types of defects affect the mark location process.

2. Marker defects

2.1. Additive defects

Penrose marker patterns with segments deliberately obscured have been fabricated on blank silicon wafers using standard resist bi-layers by ebeam in a Vistec VB6 UHR EWF and transferred to Ti/Au by lift-off. The patterns have additional defects that range in

* Corresponding author. Tel.: +44 141 330 8631.
E-mail addresses: kevin.docherty@elec.gla.ac.uk, kevinedocherty@hotmail.com (K.E. Docherty).

0167-9317/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.mee.2008.11.037

size, from 10 nm to 600 nm and in density from 1 to 10 defects per μm^2. Since these defects are part of the pattern file used to define the markers they are created in the same layer of metal. Therefore they have the same contrast and completely replace that section of the marker. Fig. 1 shows a range of markers with (a) varying density of defects and (b) varying size of defects.

2.2. Subtractive defects

A second set of marker patterns were created with sections of the pattern removed. Varying the dose used during the ebeam exposure of the markers, two effects on the markers were observed. As the dose decreased from the ideal dose-to-size for the patterns, the size of the individual elements within the pattern reduces. Further reduction of the dose caused elements of the pattern to be omitted until at the lowest doses none of the marker pattern was transferred into the lifted-off metal. This range is shown in Fig. 2.

Each of the marker patterns was written by alignment to four good Penrose markers around the corners of the written area. This allowed them to be created at a known and well-defined position. To investigate how well the markers could be used for alignment an additional alignment step was performed with the good Penrose markers, but rather than exposing a pattern an image was collected at the exact centre of the damaged marker pattern. Correlating this collected image against the pre-defined reference image allows us to get a measure of the error in the marker search induced by the damage to the marker pattern. Repeating this process at several markers of each type of damage allows us to measure the variation in the marker searches and hence to determine the achievable alignment accuracy from the standard deviation of the marker searches.

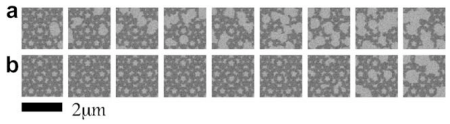


Fig. 1. Examples of the fabricated Penrose markers with added defects ranging in (a) density and (b) size.

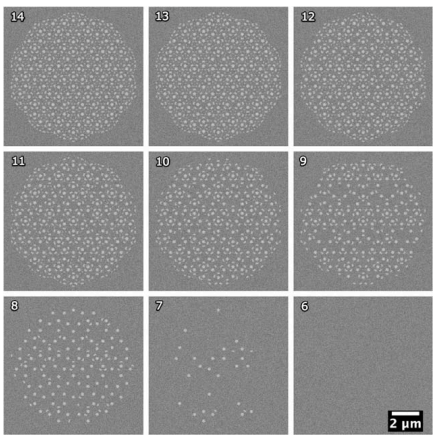


Fig. 2. Examples of the Penrose markers fabricated by varying the exposure dose demonstrating the changing size of and the omission of pattern elements as the dose reduces. Each marker in the range was numbered corresponding to the dose used and these will be referred to later.

3. Results and discussion

The performance of the different types of damaged markers has been investigated as described above. For the additive defects the imaging was performed in a Vistec EBPG5000+ ES HR and the results are shown in Fig. 3. Part (a) shows the effect that increasing the density of defects has on the accuracy with which the marker searches can be performed. It is clear to see that as the density increases so too does the standard deviation of the marker searches and therefore the accuracy of any alignment would decrease. It is interesting to note that the error only increases by ~1.5 nm as the density changes across the full range illustrated in Fig. 1a, and even in the worst-case is still less than 3 nm.

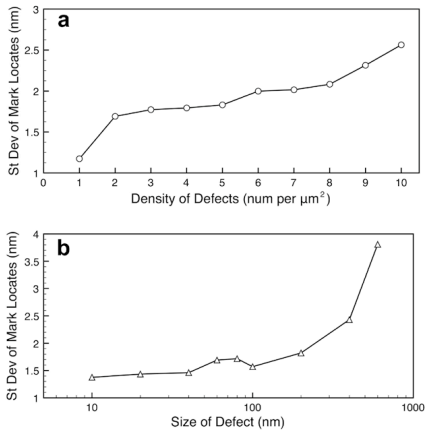


Fig. 3. Standard deviation of marker searches as (a) the number of, and, (b) the size of, the additional defects within the marker area is varied.

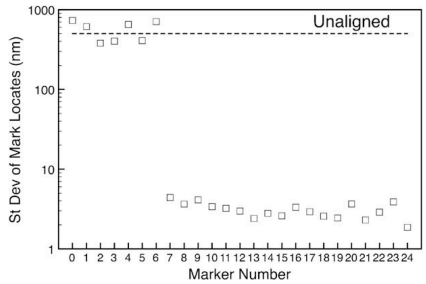


Fig. 4. Standard deviation of marker searches for the incomplete markers. Marker completeness is proportional to marker number with 0 being nothing to 24 being perfect.

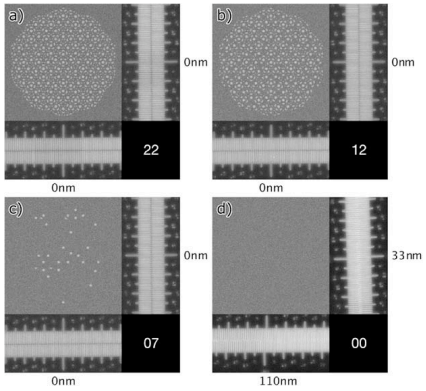


Fig. 5. SEM images of the fabricated aligned Verniers structures beside a typical marker of that dose. The marker number is written in the bottom-right corner and the alignment indicated by the Verniers is written beside them.

In Fig. 3b the standard deviation of marker searches is plotted as the size of the additional defects is increased from 10 nm to 600 nm. Again as the defect size increases it is not too surprising that the accuracy should reduce however it is interesting that the measured standard deviation remains consistent at about 1.5 nm until the defects reach 100 nm in size and then there is a large change in the measured accuracy. Again the increase is only about 2.5 nm for the worst-case of 600 nm defects and the marker search accuracy is still better than 5 nm.

The marker patterns with missing elements were imaged in a Vistec VB6 UHR EWF and we see a similar trend with more damaged markers producing larger errors and less accurate alignment. Fig. 4 shows the standard deviation of the marker search results for this type of markers. The marker numbers on the *x*-axis correspond with those shown previously on the marker images in Fig. 2. It is interesting to see that the marker search standard deviation remains roughly constant while there is still something of the marker present. However once the image becomes blank the marker search results essentially become random and there would be no alignment possible.

To prove that these marker search results equate to real alignment performance we have used sets of the underexposed markers to align and fabricate 1 nm scale Vernier structures in two halves, as a measure of the alignment possible. Four examples are shown in Fig. 5, the first (a) with a perfect marker, (b) with a good Penrose marker, the third (c) with the most damaged marker that still produced a sensible alignment, and lastly (d) with one of the markers that gave unaligned results in the marker searches. These results confirm that structures with sub-5 nm positional errors can be fabricated by aligning to these heavily damaged Penrose markers.

4. Conclusions

We have shown the high robustness of correlation-based alignment with Penrose patterns to two significant types of marker damage. With additional defects replacing significant parts of the pattern we have shown that marker searches with a standard deviation of well below 5 nm can repeatability be performed. When sections of the Penrose marker pattern are removed through underexposure we again demonstrate the high robustness of the correlation algorithm, showing sub-5 nm marker searches. Finally, fabricating aligned Vernier structures with alignment errors of less than 5 nm have been created showing the real alignment performance of damaged Penrose markers.

Acknowledgements

The authors would like to thank the staff at the James Watt Nanofabrication Centre, Glasgow, for access to the VB6 and also staff at Center of MicroNanoTechnology, EPFL, for access to their EBPG5000+. We would also like to acknowledge the funding and support received from Vistec Lithography and EPSRC which allowed this work to be performed.

References

[1] D.M. Holburn, G.A.C. Jones, H. Ahmed, Journal of Vacuum Science and Technology 19 (4) (1981) 1229–1233.
[2] E.H. Anderson, D. Ha, J.A. Liddle, Microelectronic Engineering (73–74) (2004) 74–79.
[3] V. Boegli, D.P. Kern, Journal of Vacuum Science and Technology B 8 (6) (1990) 1994–2000.
[4] A.M. Bruckstein, L. O'. Gorman, A. Orlitsky, IEEE Transactions on Information Theory 44 (7) (1998) 3156–3162.
[5] K.E. Docherty, S. Thoms, P. Dobson, J.M.R. Weaver, Microelectronic Engineering (85) (2008) 761–763.

References

[1] P. Hahmann and O. Fortagne, "50 years of electron beam lithography: Contributions from Jena (Germany)," *Microelectronic Engineering*, p. 4, 2008.

[2] D. Cumming, S. Thoms, J. M. R. Weaver, and S. Beaumont, "3 nm NiCr wires made using electron beam lithography and PMMA resist," *Microelectronic Engineering*, 1996.

[3] B. Maile, W. Henschel, H. Kurz, B. Rienks, R. Polman, and P. Kaars, "Sub-10 nm linewidth and overlay performance achieved with a fine-tuned EBPG-5000 TFE electron beam lithography system," *Jpn J Appl Phys*, vol. 39, pp. 6836–6842, 2000.

[4] A. E. Grigorescu, M. C. van der Krogt, C. W. Hagen, and P. Kruit, "10 nm lines and spaces written in HSQ, using electron beam lithography," *Microelectronic Engineering*, vol. 84, pp. 822–824, 2007.

[5] A. E. Grigorescu, M. C. van der Krogt, and C. W. Hagen, "Sub-10-nm structures written in ultra-thin HSQ resist layers using electron-beam lithography," *Proc. SPIE*, 2007.

[6] K. Yamazaki and H. Namatsu, "5-nm-order electron-beam lithography for nanodevice fabrication," *Japanese Journal of Applied Physics Part 1-Regular Papers Short Notes & Review Papers*, vol. 43, pp. 3767–3771, 2004.

[7] K. Selinidis, E. Thompson, G. Schmid, N. Stacey, J. Perez, J. Maltabes, D. Resnick, J. Yeo, H. Kim, and B. Eynon, "32 nm imprint masks using variable shape beam pattern generators," *Proc. SPIE*, vol. 7028, pp. 70 280R–1, 2009.

[8] S. Miyauchi, K. Tanaka, and J. Russ, "Automatic pattern positioning of scanning electron beam exposure," *IEEE Transactions on Electron Devices*, vol. 17, pp. 450–457, 1970.

[9] C. Ah, Y. Yun, J. Lee, H. Park, D. Ha, and W. Yun, "Fabrication of integrated nanogap electrodes by surface-catalyzed chemical deposition," *Applied Physics Letters*, vol. 88, p. 133116, 2006.

[10] N. A. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. M. Petroff, and J. R. Heath, "Ultrahigh-density nanowire lattices and circuits," *Science*, vol. 300, pp. 112–115, 2003.

[11] D. Stephani and E. Froschle, "High-precision automatic alignment procedure for vector scan e-beam lithography," *J Vac Sci Technol*, vol. 15, pp. 906–908, 1978.

[12] W. Stickel, "Method of optimizing registration signals for electron-beam microfabrication," *J Vac Sci Technol*, vol. 15, pp. 901–905, 1978.

[13] D. M. Holburn, G. A. C. Jones, and H. Ahmed, "A pattern-recognition technique using sequences of marks for registration in electron-beam lithography," *J Vac Sci Technol*, vol. 19, pp. 1229–1233, 1981.

[14] M. J. E. Golay, "Point arrays having compact, nonredundant autocorrelations," *Journal of the Optical Society of America*, vol. 61, pp. 272–273, 1971.

[15] V. Boegli and D. P. Kern, "Automatic mark detection in electron-beam nanolithography using digital image-processing and correlation," *J Vac Sci Technol B*, vol. 8, pp. 1994–2001, 1990.

[16] A. M. Bruckstein, L. O'Gorman, and A. Orlitsky, "Design of shapes for precise image registration," *Ieee Transactions on Information Theory*, vol. 44, pp. 3156–3162, 1998.

[17] Y. Chen, W. H. Huang, and X. M. Dang, "Design and analysis of two-dimensional zero-reference marks for alignment systems," *Review of Scientific Instruments*, vol. 74, pp. 3549–3553, 2003.

[18] E. H. Anderson, D. Ha, and J. A. Liddle, "Sub-pixel alignment for direct-write electron beam lithography," *Microelectronic Engineering*, vol. 73-74, pp. 74–79, 2004.

[19] B. Icard, L. Pain, J. L. Denmat, and S. Manakli, "New alignment marks for improved measurement maturity," *Proceedings of SPIE*, vol. 6792, p. 679211, 2008.

[20] A. Chang, P. Gaal, S. W. Golomb, G. Gong, T. Helleseth, and P. Kumar, "On a conjectured ideal autocorrelation sequence and a related triple-error correcting cyclic code," *Ieee Transactions on Information Theory*, vol. 46, pp. 680–687, 2000.

[21] G. Forney-Jr., "Convolutional codes I: Algebraic structure," *Ieee Transactions on Information Theory*, vol. 16, pp. 720–738, 1970.

[22] A. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Transactions on Communication Technology*, vol. 19, pp. 751–772, 1971.

[23] R. H. Dicke, "Scatter-hole cameras for X-rays and gamma rays," *Astrophysical Journal*, vol. 153, pp. L101–L106, 1968.

[24] J. P. Roques, "Fast decoding algorithm for uniformly redundant arrays," *Applied Optics*, vol. 26, pp. 3862–3865, 1987.

[25] R. L. Blake, A. J. Burek, E. E. Fenimore, and R. Puetter, "Solar X-ray photography with multiplex pin-hole camera," *Review of Scientific Instruments*, vol. 45, pp. 513–516, 1974.

[26] C. Brown, "Multiplex imaging with multiple-pinhole cameras," *Journal of Applied Physics*, vol. 45, pp. 1806–1811, 1974.

[27] E. E. Fenimore and T. M. Cannon, "Coded aperture imaging with uniformly redundant arrays," *Applied Optics*, vol. 17, pp. 337–347, 1978.

[28] E. E. Fenimore, "Coded aperture imaging - predicted performance of uniformly redundant arrays," *Applied Optics*, vol. 17, pp. 3562–3570, 1978.

[29] A. T. Moffet, "Minimum-redundancy linear arrays," *Ieee Transactions on Antennas and Propagation*, vol. Ap16, pp. 172–175, 1968.

[30] E. E. Fenimore, "Coded aperture imaging - the modulation transfer-function for uniformly redundant arrays," *Applied Optics*, vol. 19, pp. 2465–2471, 1980.

[31] A. Busboom, H. D. Schotten, and H. Elders-Boll, "Coded aperture imaging with multiple measurements," *Journal of the Optical Society of America A-Optics Image Science and Vision*, vol. 14, pp. 1058–1065, 1997.

[32] D. V. Sarwate and M. B. Pursley, "Cross-correlation properties of pseudorandom and related sequences," *Proceedings of the Ieee*, vol. 68, pp. 593–619, 1980.

[33] U. C. G. Fiebig and M. Schnell, "Correlation-properties of extended m-sequences," *Electronics Letters*, vol. 29, pp. 1753–1755, 1993.

[34] H. D. Lüke, "Perfect pseudoperiodic binary arrays," *Electronics Letters*, vol. 34, pp. 1823–1824, 1998.

[35] R. Spann, "A 2-dimensional correlation property of pseudorandom maximal length sequences," *Proceedings of the Ieee*, vol. 53, p. 2137, 1965.

[36] F. J. Macwilliams and N. J. A. Sloane, "Pseudo-random sequences and arrays," *Proceedings of the Ieee*, vol. 64, pp. 1715–1730, 1976.

[37] B. V. K. V. Kumar and L. Hassebrook, "Performance-measures for correlation filters," *Applied Optics*, vol. 29, pp. 2997–3006, 1990.

[38] L. A. Romero and F. M. Dickey, "Comparison between the peak-to-sidelobe ratio of the matched and phase-only filters," *Opt. Lett*, vol. 16, pp. 253–254, 1991.

[39] F. M. Dickey and L. A. Romero, "Dual optimality of the phase-only filter," *Opt. Lett*, vol. 14, pp. 4–5, 1989.

[40] B. V. K. V. Kumar, F. M. Dickey, and J. M. Delaurentis, "Correlation filters minimizing peak location errors," *Journal of the Optical Society of America A-Optics Image Science and Vision*, vol. 9, pp. 678–682, 1992.

[41] A. Mahalanobis, B. V. K. V. Kumar, and D. Casasent, "Minimum average correlation-energy filters," *Applied Optics*, vol. 26, pp. 3633–3640, 1987.

[42] R. Muise, A. Mahalanobis, R. Mohapatra, X. Li, D. Han, and W. Mikhael, "Constrained quadratic correlation filters for target detection," *Applied Optics*, vol. 43, pp. 304–314, 2004.

[43] A. Aviram and M. A. Ratner, "Molecular rectifiers," *Chemical Physics Letters*, vol. 29, pp. 277–283, 1974.

[44] C. Kergueris, J. Bourgoin, S. Palacin, D. Esteve, C. Urbina, M. Magoga, and C. Joachim, "Electron transport through a metal-molecule-metal junction," *Physical Review B*, vol. 59, pp. 12 505–12 513, 1999.

[45] N. Néel, J. Kröger, R. Berndt, and E. Pehlke, "Spectroscopy of an atom between two electrodes," *Physical Review B*, vol. 78, pp. 1–4, 2008.

[46] R. Smit, Y. Noat, C. Untiedt, N. Lang, M. C. van Hemert, and J. M. van Ruitenbeek, "Measurement of the conductance of a hydrogen molecule," *Nature*, vol. 419, pp. 906–909, 2002.

[47] J. van Ruitenbeek, A. Alvarez, I. Piñeyro, C. Grahmann, P. Joyez, M. H. Devoret, D. Esteve, and C. Urbina, "Adjustable nanofabricated atomic size contacts," *Review of Scientific Instruments*, vol. 67, pp. 108–111, 1996.

[48] J. Reichert, R. Ochs, D. Beckmann, H. Weber, M. Mayor, and H. van Löhneysen, "Driving current through single organic molecules," *Physical Review Letters*, vol. 88, pp. 1–4, 2002.

[49] A. Champagne, A. Pasupathy, and D. Ralph, "Mechanically-adjustable and electrically-gated single-molecule transistors," *Nano Letters*, vol. 5, pp. 305–308, 2005.

[50] C. Muller and R. de Bruyn Ouboter, "Fabrication of inherently stable and adjustable contacts of atomic size," *Journal of Applied Physics*, vol. 77, pp. 5231–5236, 1995.

[51] J. Moreland and J. Ekin, "Electron tunneling experiments using Nb-Sn "break" junctions," *Journal of Applied Physics*, vol. 58, pp. 3888–3895, 1985.

[52] M. Tsutsui, M. Taniguchi, and T. Kawai, "Fabrication of 0.5 nm electrode gaps using self-breaking technique," *Applied Physics Letters*, vol. 93, p. 163115, 2008.

[53] T. Junno, S. Carlsson, H. Xu, L. Montelius, and L. Samuelson, "Fabrication of quantum devices by ångström-level manipulation of nanoparticles with an atomic force microscope," *Applied Physics Letters*, vol. 72, pp. 548–550, 1998.

[54] C. Zhou, C. Muller, M. Deshpande, J. W. Sleight, and M. A. Reed, "Microfabrication of a mechanically controllable break junction in silicon," *Applied Physics Letters*, vol. 67, pp. 1160–1162, 1995.

[55] C. Martin, D. Ding, H. van der Zant, and J. M. van, "Lithographic mechanical break junctions for single-molecule measurements in vacuum: possibilities and limitations," *New Journal of Physics*, vol. 10, p. 065008, 2008.

[56] M. Austin and S. Y. Chou, "Fabrication of nanocontacts for molecular devices using nanoimprint lithography," *J Vac Sci Technol B*, vol. 20, pp. 665–667, 2002.

[57] A. Mahapatro, S. Ghosh, and D. Janes, "Nanometer scale electrode separation (nanogap) using electromigration at room temperature," *IEEE Transactions on Nanotechnology*, vol. 5, pp. 232–236, 2006.

[58] H. S. J. V. D. Zant, Y.-V. Kervennic, M. Poot, K. O'neill, Z. D. Groot, J. M. Thijssen, H. B. Heersche, N. Stuhr-Hansen, T. Bjørnholm, D. Vanmaekelbergh, C. A. V. Walree, and L. W. Jenneskens, "Molecular three-terminal devices: fabrication and measurements," *Faraday Discuss.*, vol. 131, pp. 347–356, 2006.

[59] F. O. Hadeed and C. Durkan, "Controlled fabrication of 1–2nm nanogaps by electromigration in gold and gold-palladium nanowires," *Applied Physics Letters*, vol. 91, p. 123120, 2007.

[60] H. Park, A. Lim, A. P. Alivisatos, J. Park, and P. L. McEuen, "Fabrication of metallic electrodes with nanometer separation by electromigration," *Applied Physics Letters*, vol. 75, pp. 301–303, 1999.

[61] M. Lambert, M. Goffman, J. Bourgoin, and P. Hesto, "Fabrication and characterization of sub-3 nm gaps for single-cluster and single-molecule experiments," *Nanotechnology*, vol. 14, pp. 772–777, 2003.

[62] P. Ho and T. Kwok, "Electromigration in metals," *Reports on Progress in Physics*, vol. 52, pp. 301–348, 1989.

[63] A. Erbe, W. Jiang, Z. Bao, D. Abusch-Magder, D. M. Tennant, E. Garfunkel, and N. Zhitenev, "Nanoscale patterning in application to materials and device structures," *J Vac Sci Technol B*, vol. 23, pp. 3132–3137, 2005.

[64] A. Bezryadin and C. Dekker, "Nanofabrication of electrodes with sub-5 nm spacing for transport experiments on single molecules and metal clusters," *J Vac Sci Technol B*, vol. 15, pp. 793–799, 1997.

[65] G. Gazzadi and S. Frabboni, "Fabrication of 5 nm gap pillar electrodes by electron-beam Pt deposition," *J Vac Sci Technol B*, vol. 23, pp. L1–L3, 2005.

[66] E. Soldatov and E. Ovchenkov, "Formation of nanogaps in electrochemically deposited films of metals by the stress relaxation method," *Technical Physics*, vol. 53, pp. 1606–1608, 2008.

[67] J. Kroger, N. Neel, and L. Limot, "Contact to single atoms and molecules with the tip of a scanning tunnelling microscope," *Journal of Physics: Condensed Matter*, vol. 20, p. 223001, 2008.

[68] R. D. Piner, ""Dip-pen" nanolithography," *Science*, vol. 283, pp. 661–663, 1999.

[69] R. Krahne, A. Yacoby, H. Shtrikman, I. Bar-Joseph, T. Dadosh, and J. Sperling, "Fabrication of nanoscale gaps in integrated circuits," *Applied Physics Letters*, vol. 81, pp. 730–732, 2002.

[70] T. Kizuka, "Atomic configuration and mechanical and electrical properties of stable gold wires of single-atom width," *Physical Review B*, vol. 77, pp. 1–11, 2008.

[71] N. Agraït, A. Yeyati, and J. van Ruitenbeek, "Quantum properties of atomic-sized conductors," *Physics Reports*, vol. 377, pp. 81–279, 2003.

[72] D. Bowler, P. Anderson, P. Lee, and M. Randeria, "Atomic-scale nanowires: physical and electronic structure," *Journal of Physics: Condensed Matter*, vol. 16, pp. R721–R754, 2004.

[73] C. Muller, J. van Ruitenbeek, and L. D. Jongh, "Conductance and supercurrent discontinuities in atomic-scale metallic constrictions of variable width," *Physical Review Letters*, vol. 69, pp. 140–143, 1992.

[74] W. Chao, J. Kima, S. Rekawa, P. Fischer, and E. Anderson, "HSQ double patterning process for 12 nm resolution X-ray zone plates," *Lawrence Berkeley National Laboratory*, 2009.

[75] M. Kameyama, M. McCallum, and S. Owa, "Evolution of wavelength shrinkage in lithography," *Proceedings of SPIE*, vol. 7281, pp. 2–6, 2009.

[76] I. Hanyu, S. Asai, K. Kosemura, H. Ito, M. Nunokawa, and M. Abe, "New phase-shifting mask with highly transparent SiO_2 phase shifters," *Proceedings of SPIE*, vol. 1264, p. 167, 1990.

[77] D. Levenson, "What is a phase-shifting mask?" *Proceedings of SPIE*, vol. 1496, p. 20, 1991.

[78] B. Lin, "Phase-shifting and other challenges in optical mask technology," *Proceedings of SPIE*, vol. 1496, p. 54, 1991.

[79] C. Eriksson, B. Wiklund, and B. Astrand, "A thermal heated field emitter gun for e-beam exposure," *Phys Scripta*, vol. 24, pp. 477–481, 1981.

[80] L. Swanson, "Comparative study of the zirconiated and built-up W thermal-field cathode," *J Vac Sci Technol*, 1975.

[81] L. Pain, M. Jurdit, J. Todeschini, S. Manakli, B. Icard, B. Minghetti, G. Bervin, A. Beverina, F. Leverd, and M. Broekaart, "Electron beam direct write lithography flexibility for asic manufacturing: an opportunity for cost reduction (keynote paper)," *Proceedings of SPIE*, vol. 5751, p. 35, 2005.

[82] L. Pain, B. Icard, S. Manakli, J. Todeschini, B. Minghetti, V. Wang, and D. Henry, "Transitioning of direct e-beam write technology from research and development into production flow," *Microelectronic Engineering*, vol. 83, pp. 749–753, 2006.

[83] J. Todeschini, L. Pain, S. Manakli, B. Icard, V. Dejonghe, B. Minghetti, M. Jurdit, D. Henry, and V. Wang, "Electron beam direct write process development for sub 45nm cmos manufacturing," *Proceedings of SPIE*, vol. 5753, p. 408, 2005.

[84] S. Thoms, "BELLE - beamwriter exposure layout for lithographic engineers," *private communication*, 2002.

[85] ——, "Runjob - VB6 control software for a multi-user environment," *private communication*, 2009.

[86] E. Reichmanis and A. E. Novembre, "Lithographic resist materials chemistry," *Annu Rev Mater Sci*, vol. 23, pp. 11–43, 1993.

[87] M. Hatzakis, "Electron resists for microcircuit and mask production," *Journal of The Electrochemical Society*, vol. 116, pp. 1033–1037, 1969.

[88] A. Hoole, M. Welland, and A. Broers, "Negative PMMA as a high-resolution resist-the limits and possibilities," *Semiconductor Science and Technology*, 1997.

[89] I. Ziler, J. E. F. Frost, V. ChabasseurMolyneux, C. J. B. Ford, and M. Pepper, "Crosslinked PMMA as a high-resolution negative resist for electron beam lithography and applications for physics of low-dimensional structures," *Semiconductor Science and Technology*, vol. 11, pp. 1235–1238, 1996.

[90] R. Tiron, L. Mollard, O. Louveau, and E. Lajoinie, "Ultrahigh-resolution pattern using electron-beam lithography HF wet etching," *J Vac Sci Technol B*, 2007.

[91] J. K. W. Yang and K. K. Berggren, "Using high-contrast salty development of hydrogen silsesquioxane for sub-10-nm half-pitch lithography," *J Vac Sci Technol B*, vol. 25, pp. 2025–2029, 2007.

[92] R. P. Howson, "The reactive sputtering of oxides and nitrides," *Pure and Applied Chemistry*, vol. 66, pp. 1311–1311, 1994.

[93] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing.* Numerical Recipes, 1986.

[94] J. Goldstein, D. Newbury, D. Joy, C. Lyman, P. Echlin, E. Lifshin, L. Sawyer, and J. Michael, *Scanning Electron Microscopy and X-Ray Microanalysis.* Springer, 2003.

[95] E. D. Wolf, P. J. Coane, and F. S. Ozdemir, "Composition and detection of alignment marks for electron-beam lithography," *J Vac Sci Technol*, vol. 12, pp. 1266–1270, 1975.

[96] A. D. Wilson, T. H. P. Chang, and A. Kern, "Experimental scanning electron-beam automatic registration system," *J Vac Sci Technol*, vol. 12, pp. 1240–1245, 1975.

[97] G. L. Turin, "An introductions to matched-filters," *Ire T Inform Theor*, vol. 6, pp. 311–329, 1960.

[98] D. Erdogmus, R. Agrawal, and J. Principe, "A mutual information extension to the matched filter," *Signal Process*, vol. 85, pp. 927–935, 2005.

[99] P. Refregier, "Optimal trade-off filters for noise robustness, sharpness of the correlation peak, and horner efficiency," *Opt. Lett*, vol. 16, pp. 829–831, 1991.

[100] W. M. Siebert, "A radar detection philosophy," *Ire Transactions on Information Theory*, vol. 2, pp. 204–221, 1956.

[101] R. N. Bracewell, *The Fourier transform and its applications*. New York ; London : McGraw-Hill, 1986.

[102] S. D. Barthelmy, "The burst alert telescope (BAT) on the swift MIDEX mission," *SPIE*, vol. 4140, p. 50, 2000.

[103] Bazzano, Amoretti, L. Padula, Natalucci, Soggiu, Ubertini, and Waldron, "MART-LIME: A high energy imageing and spectroscopic experiment on board spectrum-X-gamma," *SPIE Proceedings on Advances in Multilayer and Graxing Incidence X-Ray/EUV/FUV Optics*, vol. 2279, p. 446, 1994.

[104] C. Zhang and S. Zhang, "A super-high angular resolution principle for coded-mask x-ray imaging beyond the diffraction limit of a single pinhole," *Research in Astronomy and Astrophysics*, vol. 9, pp. 333–340, 2009.

[105] A. Busboom and H. D. Luke, "Hexagonal binary arrays with perfect correlation," *Applied Optics*, vol. 40, pp. 3894–3900, 2001.

[106] A. Busboom, H. Elders-Boll, and H. D. Schotten, "Uniformly redundant arrays," *Exp Astron*, vol. 8, pp. 97–123, 1998.

[107] S. R. Gottesman and E. E. Fenimore, "New family of binary arrays for coded aperture imaging," *Applied Optics*, vol. 28, pp. 4344–4352, 1989.

[108] J. Braga, F. D'Amico, T. Villela, J. Mejia, R. A. Fonseca, and E. Rinke, "Development of the imaging system of the balloon-borne gamma-ray telescope mascara codificada (MASCO)," *Review of Scientific Instruments*, vol. 73, pp. 3619–3628, 2002.

[109] A. Busboom, "Construction of pseudo-noise arrays from quadratic residues," *Signal Process*, vol. 72, pp. 33–38, 1999.

[110] R. Barker, "Group synchronizing of binary digital systems in communications theory," *Acdemic Press, New York*, pp. 273–287, 1953.

[111] W. Silver, "Article of manufacture bearing a universal alignment target," *United States Patent 6671049*, 2003.

[112] R. Penrose, *The emperor's new mind : concerning computers, minds and the laws of physics*. Oxford University Press, 1989.

[113] W. Rasband. (2006) ImageJ - image processing and analysis in java.

[114] T. H. P. Chang, "Proximity effect in electron-beam lithography," *J Vac Sci Technol*, vol. 12, pp. 1271–1275, 1975.

[115] Vistec Lithography Ltd., "Vectorbeam series – EMMA software manuals – version v2008.01.385," *Cambridge, England Vistec Lithography Ltd*, 2009.

[116] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *The Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[117] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, pp. 431–441, 1963.

[118] M. Lourakis. (2004) levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++.

[119] D. S. Macintyre and S. Thoms, "Nanometre scale overlay and stitch metrology using an optical microscope," *Microelectronic Engineering*, vol. 83, p. 1051, 2006.

[120] M. Postek, A. Vladar, S. N. Jones, and W. Keery, "Interlaboratory study on the lithographically produced scanning electron microscope magnification standard prototype," *Journal of Research of the National Institute of Standards and Technology*, vol. 98, pp. 447–467, 1993.

[121] K. A. Lister, "Use of EBPG5000+ ES HR for measuring correlation performance when using damaged markers," *private communication*, 2008.

[122] K. G. Kemp, "Lithographic method using double exposure techniques, mask position shifting and light phase shifting," *U S Patent 5308741*, 1994.

[123] E. Anderson and W. Chao, "Double exposure makes dense high-resolution diffractive optics," *SPIE Newsroom*, 2007.

[124] M. A. Reed, C. Zhou, C. J. Muller, T. P. Burgin, and J. M. Tour, "Conductance of a molecular junction," *Science*, vol. 278, pp. 252–254, 1997.

[125] C. Zhou, C. J. Muller, M. Deshpande, J. Sleight, and M. A. Reed, "Microfabrication of a mechanically controllable break junction in silicon," *Applied Physics Letters*, vol. 67, pp. 1160–1162, 1995.

[126] G. Leatherman, E. Durantini, D. Gust, and T. Moore, "Carotene as a molecular wire: Conducting atomic force microscopy," *Journal of Physical Chemistry B*, vol. 103, pp. 4006–4010, 1999.

[127] P. Steinmann and J. M. R. Weaver, "Nanometer-scale gaps between metallic electrodes fabricated using a statistical alignment technique," *Applied Physics Letters*, vol. 86, p. 063104, 2005.

[128] X. Li, X. Cao, H. Zhou, C. Wilkinson, S. Thoms, D. Macintyre, M. Holland, and I. Thayne, "30 nm tungsten gates etched by a low damage ICP etching for the fabrication of compound semiconductor transistors," *Microelectronic Engineering*, vol. 83, pp. 1152–1154, 2006.

[129] K. R. Williams, K. Gupta, and M. Wasilik, "Etch rates for micromachining processing-part II," *Microelectromechanical Systems, Journal of*, vol. 12, pp. 761–778, 2003.

[130] Y. I. Gold and A. Goldenshtein, "Sem image sharpening by reversing the effect of a nonideal beam spot," *Metrology, Inspection, and Process Control for Microlithography XII*, vol. 3332, pp. 620–624, 1998.

[131] J. A. Liddle, P. Naulleau, and G. M. Schmid, "Probe shape measurement in an electron beam lithography system," *J Vac Sci Technol B*, vol. 22, pp. 2897–2901, 2004.

[132] K. Yamazaki and H. Namatsu, "Electron-beam diameter measurement using a knife edge with a visor for scattering electrons," *Jpn J Appl Phys 2*, vol. 42, pp. L491–L493, 2003.

[133] A. Goldenshtein, Y. I. Gold, and H. Chayet, "Measuring the size and intensity distribution of sem beam spot," *Proc. SPIE*, vol. 3332, p. 132, 2003.